University of Bath

PHD

On modelling and simulation of plucked string instruments

Shabana, Wafaa Rezk

Award date:
2002

Awarding institution:
University of Bath

Link to publication

Alternative formats
If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

Copyright of this thesis rests with the author. Access is subject to the above licence, if given. If no licence is specified above, original content in this thesis is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC-ND 4.0) Licence (https://creativecommons.org/licenses/by-nc-nd/4.0/). Any third-party copyright material present remains the property of its respective owner(s) and is licensed under its existing terms.

Take down policy
If you consider content within Bath's Research Portal to be in breach of UK law, please contact: openaccess@bath.ac.uk with the details. Your claim will be investigated and, where appropriate, the item will be removed from public view as soon as possible.

Download date: 05. May. 2023

On Modelling and Simulation of Plucked String Instruments

submitted by

Wafaa Rezk Shabana

for the degree of Doctor of Philosophy

of the

University of Bath

2002

COPYRIGHT

Attention is drawn to the fact that copyright of this thesis rests with its author. This copy of the thesis has been supplied on the condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Signature of Author .....Wafaa Shabana.....

Wafaa Rezk Shabana

UMI Number: U145902

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

UMI U145902
Published by ProQuest LLC 2013. Copyright in the Dissertation held by the Author.
Microform Edition © ProQuest LLC.
All rights reserved. This work is protected against
unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

UNIVERSITY OF PLYMOUTH
LIBRARY
70 27 MAY 2002
Ph.D.

Acknowledgments

I would like to express my sincere thanks to my supervisor Prof. John Fitch for his constant encouragement, endless support, fruitful discussions and above all his belief and confidence in me. Thank you for being there for me all the time. I would also like to thank Dr. Julian Padget for his help, patience and kindness. I would also like to express my gratitude to some other people who helped me in one way or another. Those are my colleagues whom I shared the office 1W2.26 for quite some time. They have been wonderful, kind, supportive and helpful. I wish you all the best.

My grateful thanks, also, are due to my husband who introduced me to the wonderful world of Kalman filter. His deep discussions and invaluable comments have helped me greatly throughout the course of my study. I am also very grateful to my two lovely boys Ayman and Ahmed for everything they have given me. You guys keep me going. Simply you are the sound of life. My overwhelming gratitude is kept for a very special person, my mum without whose love, care, support, and prayers I would never be in this position. May God bless you all.

Finally special thanks are due to the Egyptian Government and the Educational and Cultural Bureau in London for their financial support and help.

Abstract

During the last two decades physical modelling of real musical instruments has gained popularity as a tool for sound synthesis and computer music. The term *physical modelling* simply refers to the simulation of the sound-production mechanism of a real musical instrument. The instrument under study here is the classic acoustic guitar.

The modelling process starts by studying the vibrating string as it is the main source of vibration in any stringed musical instrument. The ideal vibrating string satisfies the one-dimensional (1D) wave equation which can be modelled accurately by using the *digital waveguide* techniques. Based on the principle of *commuted waveguide* and the extensions of the *Karplus-Strong* (KS) algorithm, a guitar model is designed.

Once a physically relevant model of the instrument is constructed, an analysis-based synthesis scheme is developed for the estimation of the model parameters as well as the excitation signals. Starting with a real guitar tone, the estimation process involves the following major steps:

— Estimation of the fundamental frequency, which in turn determines the nominal delay of the string model. A new pitch detection algorithm based on the *Dyadic Wavelet Transform* is utilized for this purpose. The analysis wavelet is a quadratic spline wavelet. The construction of this wavelet is explained in detail. The performance of other wavelets has been investigated for comparison. Furthermore a comparative study between the proposed algorithm and the well-known autocorrelation function is presented.

— Based on sinusoidal modelling, the estimation of the recorded signal features such as frequency, amplitude, and phase is the next step in the analysis. A new Kalman filtering-based (EKF) technique is proposed for this task. Different models have been tested and the results of this algorithm are compared to those obtained by the more conventional Short-Time Fourier Transform (STFT) based analysis.

Once the model parameters are extracted, the excitation signals are estimated. Finally the re-synthesized signal is compared to the original and some error-analysis tests are performed on the EKF-residual signal.

The proposed new techniques are proven to be accurate, robust to noise, fast, and have the potential of real-time implementation.

Keywords: physical modelling, pitch detection, Wavelet transform, Kalman filter, Short-Time Fourier Transform, partials tracking.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Digital sound synthesis has become a very active field in Musical Acoustics and Computer Music. There are different techniques for sound synthesis such as additive synthesis, subtractive synthesis, nonlinear synthesis and physical modelling synthesis. Physical modelling, in particular, has gained popularity during the last two decades. Unlike the other sound synthesis techniques, the physical modelling approach simulates the properties of the sound source rather than the properties of the sound itself. The term *physical modelling* simply refers to the simulation of the sound production mechanism of a certain musical instrument. This is achieved by studying the physics and mathematics that govern and control the behaviour of the instrument under study.

This chapter is organised as follows : section 1 presents an overview of the various physical modelling techniques. Some of the advantages and disadvantages of each technique are briefly described. In section 2 the construction of the acoustic guitar, as the instrument under study in this current work, is briefly considered. Finally the thesis outline is presented in section 3.

1.1 Physical Modelling Techniques

Sound synthesis by using simulated physical models has gained popularity as it offers the musician simpler tools for controlling and producing new and traditional musical sounds. Most often models are used to understand the physical phenomena of the musical instrument and how it works. Sometimes the physical-model techniques are used to develop a model-based sound synthesizer.

In [62] Valimaki and Takala divide physical modelling techniques into five basic categories. These categories are:

- Source-filter modelling.
- Numerical solving of partial differential equations.

- Mechanical Models.
- Modal synthesis.
- Waveguide synthesis.

These approaches are summarised briefly in the following.

1. Source-filter modelling: this technique has been used for synthesizing speech and singing voices. In this case the system under study is decomposed into two subsystems, namely, the "excitation" and the "resonator". For example the speech production mechanism where the interaction between the vocal cords and the vocal tract is modelled as a feed-forward system consisting of a source and a time-varying filter. In some musical instruments it is easy to separate the source and the filter but most musical instruments are more complicated systems than just a combination of two subsystems; for instance wind instruments.

2. Numerical solving of partial differential equations: the basic idea of this method is to approximate numerically the wave equation that governs the vibration of the object at a finite set of points. Although the finite difference method is accurate in reproducing the original waveform if the model parameters are correct, it is computationally very intensive. This technique has been applied to string instruments such as the guitar, the piano and the violin including the interaction between the player and the instrument.

3. Mechanical synthesis: in this case the acoustical system under study is modelled using simple mechanical elements such as masses, springs, and dampers. A system called CORDIS was the first to simulate a musical instrument as a collection of point masses that have certain elasticity and frictional characteristics [11]. These point masses are connected together through dampers and springs. A major simplification can be made by

considering each element as being one-dimensional. Although this method is implemented in real-time sound synthesis very successfully, again it is very intensive computationally.

4. Modal synthesis: this technique simulates the vibrating system as a collection of subsystems. These subsystems could be strings, acoustic tubes, bridges, and instrument bodies. The subsystems are connected between different access points and interact together. Despite the generality of this technique, its complexity increases rapidly with the complexity of the underlying structure.

5. Waveguide synthesis: this is based on the analytical solution of the differential equation that governs the propagation of waves in a medium. This method is very efficient in simulating not only the 1D-vibratory systems but also 2D-systems and 3D-systems. It turned out to be the most important method of all physical modelling methods. We will consider the digital waveguide synthesis of string instruments, in particular the acoustic guitar, in more detail in chapter 2.

A very thorough study of the evaluation of the various sound synthesis techniques is given in [56].

1.2 Construction of the Acoustic Guitar

All string instruments have the same fundamental functional features. They all have a string-like structure as the primary source of vibration. When the string is excited by the player (plucked, bowed, or struck) it vibrates. Mathematically it can be represented by the one-dimensional wave equation. The strings are coupled to a resonating body for sound radiation, acoustic amplification, and colouring of the sound. The strings can also interact with each other via the bridge and the

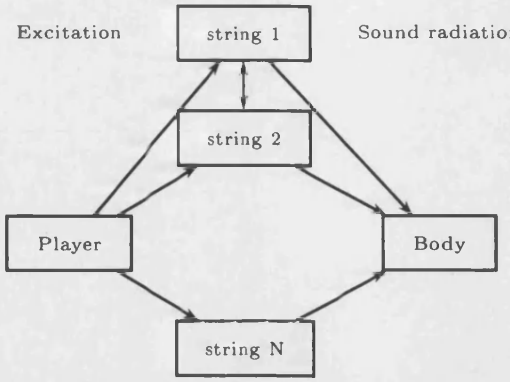

Figure 1.1: A model of a plucked-string instrument

nut [41]. A model showing the main elements and the inter-couplings of a typical plucked string instrument is shown in figure (1.1) (after Karjalainen et al. [59]).

The physics-based model used in this study is of the acoustic guitar as an example of a string instrument. The acoustic guitar has been studied extensively in the literature [22], [55] and [15].

The acoustic guitar features six strings each of which is about 64 cm in length. The material of the strings is either nylon (as in the classical style) or steel (as in folk and jazz). The strings are coupled to a resonating body through the bridge which plays a very important role. The bridge is usually made of wood (ebony). Traditionally it is glued directly to the top plate of the guitar body (sound-board). The guitar body is made mainly from wood. It consists of the top plate, the back plate, and the sides (ribs). The top plate or the sound board is considered as the most important part of the guitar as it determines the quality and character of the instrument's sound. Both the back plate and the ribs are made from the same wood.

The strings of the guitar run above the finger-board which is attached to the neck. The finger-board is made from ebony and runs along the front of the sound-board and is glued to the neck. Frets are made from metal and are mounted to the finger-board. The distance between any two frets is 1/17.817 times the shorter

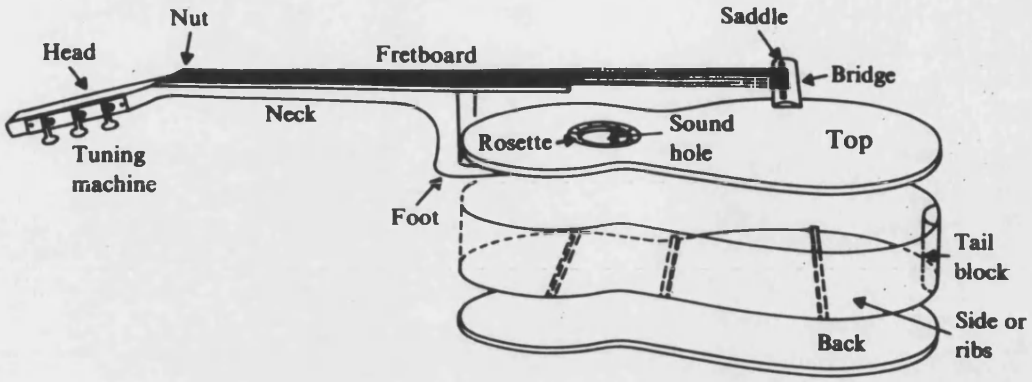

Figure 1.2: The various parts of the acoustic guitar

of the two distances to the bridge. The neck of the guitar varies in its design and shape but it is usually made from hard wood. The tuning machine is suited at the head of the guitar and it is used to tune the strings by adjusting the tension of each string. Figure (1.2) shows the various parts of the acoustic guitar [9]. A physically relevant model of the acoustic guitar is constructed in chapter 2.

1.3 Thesis Outline

Once a physics-based model is designed, it has to be calibrated to a certain instrument which is the objective of the analysis process.

Our goal in this thesis is to introduce new analysis tools for the analysis and synthesis of the guitar tones. Firstly we use a wavelet-based algorithm for pitch detection and estimation. This algorithm uses a quadratic spline wavelet as the analysis wavelet. The performance of some other wavelets is evaluated as well. Secondly we use a Kalman-based algorithm for partials tracking in guitar signals. The advantages and disadvantages of the new techniques are explained. Furthermore the results obtained by these tools are compared with the ones

obtained by the conventional methods for justification and validation.

This thesis is organised as follows: chapter 2 presents the guitar model considered in this current study. This model is designed by utilising the digital waveguide techniques and the commuted waveguide principle. In chapter 3 we propose an algorithm based on the Dyadic Wavelet Transform for estimating the pitch period. A comparative study between the proposed algorithm and the well-known autocorrelation function is also presented in this chapter. In chapter 4 the analysis process started in chapter 3 continues by estimating the decay rates of the different partials present in the guitar signal. An algorithm based on the Extended Kalman Filter (EKF) theory is utilised for this task. Various techniques for optimizing the performance of the proposed algorithm are described. The results of the algorithm are compared to those obtained by the Short-Time Fourier Transform analysis. In chapter 5 the estimation process of the underlying guitar model is explained. This involves the estimation of the string model parameters, the body resonators' parameters and the excitation signals. In this chapter a comparative study between the synthesized signal and the original one is also presented. This is done both in time and frequency domains. Moreover some error-analysis tests are performed on the residual (noise) signal of the EKF-algorithm to assess its accuracy. Chapter 6 presents the conclusions that summarize the performance of the proposed new algorithms. Finally the thesis summary and future work are presented in chapter 7.

Chapter 2

Digital Waveguide Modelling of the Acoustic Guitar

2.1 Introduction

In the previous chapter an overview of digital sound synthesis techniques is presented with emphasis on Physical Modelling. In this chapter the principles of digital waveguides as well as the model of the acoustic guitar considered in this work are presented. The digital waveguide model is based on the analytical solution of the differential equation that governs propagation of waves in a medium. This method has proven to be efficient in simulating one-dimensional vibratory systems as well as two and three-dimensional acoustic systems. The principles of digital waveguides have been developed by Smith [47], [49], [50], and [51].

Modelling the acoustic guitar can be divided into three functionally separate systems: the excitation, the strings and the body. The vibratory motion of the string, as the main resonant system in any stringed musical instrument, is presented in section 2. In section 3 principles of the one-dimensional waveguide technique of strings is described. The advantages of this technique are also presented in this section. The chapter continues by describing the well-known Karplus-Strong algorithm (KS-algorithm) for simulating plucked string tones. Advantages and disadvantages of this basic algorithm are presented as well as its extensions in section 4. Section 5 concerns with the commuted waveguide principle and body factoring technique. Finally the guitar model, based on the digital waveguide technique and extensions of KS-algorithm, is described. Further extensions and other nonlinearities are also considered.

2.2 The Vibrating String

This section describes the wave equation that governs the vibratory motion of a string and its general solution by travelling waves. It continues by describing rigidly terminated strings as well as the behaviour of real strings.

2.2.1 Ideal and Lossless Vibrating String

Strings are the main resonant system in many musical instruments. When an ideal and lossless string is set into motion (i.e by plucking), it vibrates according to Newton's second law [33]:

$$\text{Force} = \text{Mass} * \text{Acceleration}$$

The vibratory motion of the string is expressed mathematically by the one-dimensional wave equation given by:

$$Ty'' = \epsilon\ddot{y} \tag{2.1}$$

where

T – is the applied tension,

ϵ – is mass per unit length,

$\ddot{y} = \frac{\partial^2 y(x,t)}{\partial t^2}$ – is acceleration, and

$y'' = \frac{\partial^2 y(x,t)}{\partial x^2}$ – is curvature.

Equation (2.1) describes the transverse vibrations on the string, where the relevant restoring force (per unit length) is given by the applied tension times the curvature of the string. This restoring force is balanced at any time by the product of the transverse acceleration and the mass density (inertial force per unit length).

D'Alembert's solution of the wave equation is given by [16], [17], [9]:

$$y(x,t) = y_r(x - ct) + y_l(x + ct) \tag{2.2}$$

where $c = \sqrt{T/\epsilon}$ is the sound velocity, y_r and y_l are two arbitrary, twice differentiable functions in both x and t. These functions can be determined from the initial conditions. Equation (2.2) can be interpreted physically as "the displacement of a vibrating ideal and lossless string at any time is the sum of two waves

travelling in opposite directions with speed c". Note that any perfectly elastic medium, which is displaced along one dimension, can be modelled with equation (2.2) using the appropriate variables; for instance an organ pipe or an air column of a clarinet.

2.2.2 Rigid Terminations

Rigid termination is the simplest case of terminating a string. A rigidly terminated string can not move at all at the termination ends. For an ideal string of finite length L, that is rigidly terminated, the boundary conditions are:

$$\begin{aligned} y(0,t) &= 0 \\ y(L,t) &= 0 \end{aligned}$$

Consequently the travelling waves will be reflected at the ends with inverted polarity forming standing waves, i.e.

$$y_l(ct) = -y_r(-ct)$$

$$y_l(L+ct) = -y_r(L-ct)$$

Therefore the vibratory motion of the string is periodic with period $2L/c$. It is considered as a combination of several modes of vibration. It is worth mentioning here that, theoretically, harmonics of the string that have a node at the excitation point will be absent, i.e. if the string is excited at its mid point, then all the even harmonics will be absent [9] [34].

2.2.3 Lossy and Dispersive One-dimensional Wave Equation

In real strings physical phenomena that account for attenuating the vibratory motion are always present. Energies are lost via the string terminations and due

to internal friction and air damping [9]. These losses damp the almost periodic vibration of the string and add to the wave equation a term that is proportional to $\dot{y}$.

The lossy, one-dimensional wave equation is given by:

$$Ty'' = \epsilon\,\ddot{y} + \mu\,\dot{y} \tag{2.3}$$

where μ is a proportionality constant. The solution to this modified equation can be written as:

$$y(x,t) = e^{-(\frac{\mu}{2\epsilon})\frac{x}{c}}\,y_r(x-ct) + e^{(\frac{\mu}{2\epsilon})\frac{x}{c}}\,y_l(x+ct) \tag{2.4}$$

that is both the left-going and the right-going components decay exponentially in their respective direction of travel [49].

Stiffness is another property of a real string. It introduces a restoring force that is proportional to the fourth derivative of the string displacement. This results in increasing the wave propagation speed with frequency, i.e high frequencies propagate much faster than low ones. Hence the travelling waves are no longer static waves propagated with speed c and expressed as functions of $(x+ct)$ and $(x-ct)$. They will disperse as they propagate along the string (evolve with time). A stiff string is considered as an intermediate stage between an ideal string and an ideal bar. In almost all real strings losses increase with frequency [9].

2.3 The One-dimensional Digital Waveguide

In this section the principle of the digital waveguide model is described. It is mainly based on the idea of sampling the general solution (2.2) of the wave equation in both time and space. The efficiency of this model is that it can be implemented using digital filters and delay lines. Also the lossy and dispersive waveguide model of a real string is addressed. This section continues by describing the consolidation of all losses and dispersion to form a single-loop formulation.

2.3.1 The One-dimensional, Lossless Waveguide Model

Waveguide modelling is an efficient method for simulating one-dimensional resonators such as a vibrating string, or a thin bar as well as wind and string instruments. Also this technique has proven to be useful in simulating two and three-dimensional vibrating systems like plates and drums [51].

The main idea of a digital waveguide model is to sample the solution of the wave equation in time and space in order to obtain a discrete-time model. By assuming that the travelling waveforms of the wave equation are band-limited signals to half the sampling frequency (Nyquist limit), the travelling waves can be sampled at intervals of T seconds, corresponding to a sampling frequency of $f_s = 1/T$ samples per second. Sampling is carried out by the change of variables:

$$x \rightarrow x_n = nX, \quad t \rightarrow t_m = mT \tag{2.5}$$

for $n, m = 0, 1, 2, \ldots$, where $X = cT$ is the spatial sampling interval.

On substituting into the solution of the wave equation, equation (2.2), yields

$$y(n, m) = y^+(n - m) + y^-(n + m) \tag{2.6}$$

where

$$y^+(n - m) = y_r((n - m)T)$$
$$y^-(n + m) = y_l((n + m)T)$$

This equation can be interpreted as a bi-directional delay line where the waveform $y^+(n - m)$ travels to the right and $y^-(n + m)$ travels to the left. Amplitude of the signal at any point is obtained by summing values of the two components at that point. The block diagram of the one-dimensional digital waveguide is shown in figure (2.1). In figure (2.1) the output is taken at two arbitrary points $x = 0$ and $x = mX$.

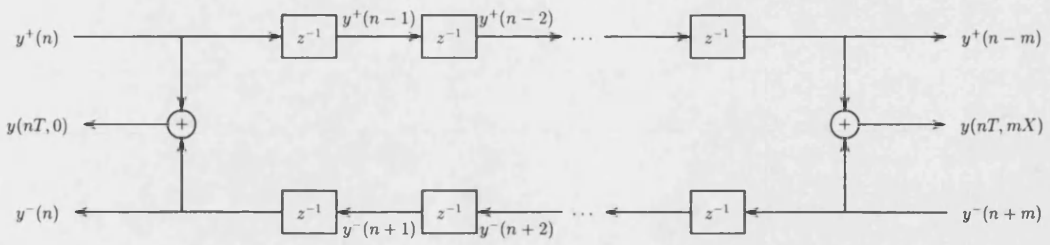

Figure 2.1: One-dimensional, ideal and lossless digital waveguide

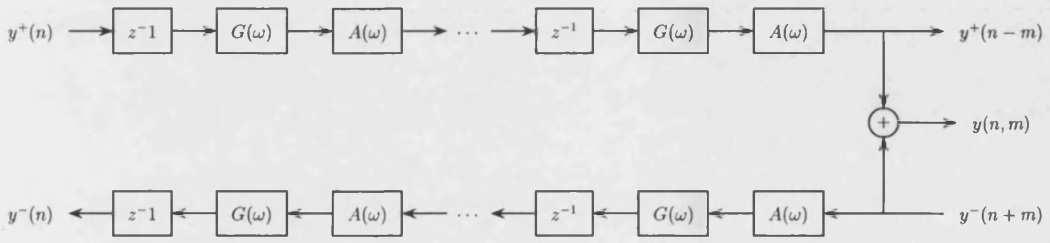

Figure 2.2: Lossy and dispersive 1D-digital waveguide

2.3.2 Lossy and Dispersive Waveguide Model

Losses and dispersion of a real string can be implemented in the digital waveguide by inserting digital filters at the sampling points to simulate the damping effect of the energy of a real string.

Frequency-dependent losses in a real string cause exponential decay of the travelling waves. This effect is simulated in the digital waveguide model by multiplying the travelling waves at each time step by a frequency-dependent constant. Also stiffness of a real string, which causes dispersion, can be taken into account in the digital waveguide model by using an all-pass filter. Such a filter has the effect of non-uniform delay of the travelling waves [49]. The block diagram of a lossy and dispersive digital waveguide is shown in figure (2.2) where $G(\omega)$ accounts for frequency-dependent factors (damping) and $A(\omega)$ accounts for frequency-dependent delay (dispersion).

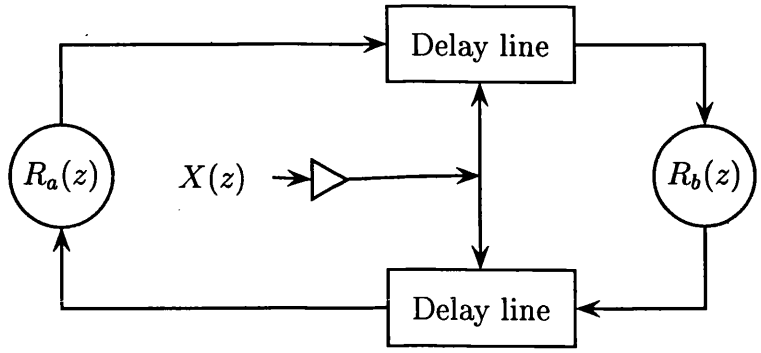

Figure 2.3: Waveguide model for a terminated real string

2.3.3 Single-delay Line Loop Formulation

Since the string is considered as a linear and time-invariant system, all the losses and other non-linearities can be collected together at a minimum number of points. This simplification results in vast reduction of computational complexity in digital waveguide implementation. Figure (2.3) shows the digital waveguide model of a real, finite length string with rigid terminations [61]. Both $R_b(z)$ and $R_a(z)$ are reflection filters that produce inversion and slightly frequency-dependent damping at the termination point (bridge) and the corresponding fret respectively. In 1998 Karjalanien et al. [61], have proposed the formulation of a single-delay loop model that is equivalent to the bi-directional model shown in figure (2.3). In this section we describe the single-delay loop model for an acoustic guitar with a transversal bridge force as its output. The model in figure (2.3) can be modified as shown in figure (2.4).

$H_{A,B}(z)$ represents the transfer function from point A to point B in the model, $F(z)$ is the transverse force at the bridge (the string model output), $Z(z)$ is the bridge impedance and $I(z)$ is the transfer function that represents the discrete time approximation of the continuous-time integration operator. The excitation signal $X(z)$ is divided into two equal parts $X_1(z)$, $X_2(z)$ that are fed into the

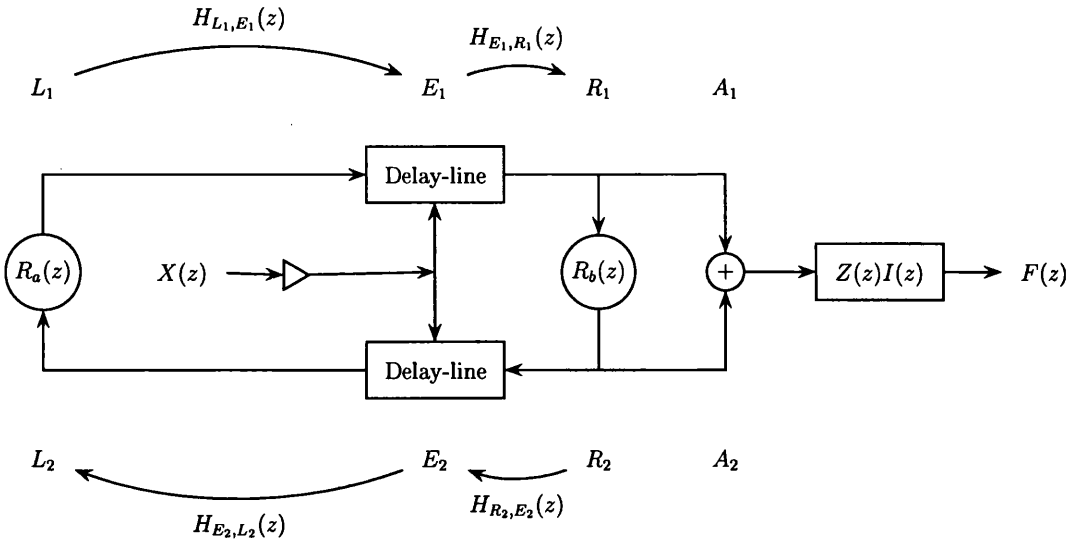

Figure 2.4: Bi-directional waveguide model with force output at the bridge

delay lines at points E_1 and E_2 respectively. An equivalent excitation signal $X_{eq,E1}$ at point E_1 is given by

$$\begin{aligned} X_{eq,E_1} &= X_2(z)H_{E_2,L_2}(z)R_a(z)H_{L_1,E_1} + X_1(z) \\ &= \frac{X(z)}{2}(1 + H_{E_2,E_1}(z)) \end{aligned} \tag{2.7}$$

where

$$H_{E_2,E_1}(z) = H_{E_2,L_2}(z)R_a(z)H_{L_1,E_1}(z)$$

is the transfer function from point E_2 to point E_1.

Given that both $A_1(z)$ and $A_2(z)$ represent right-going and left-going travelling acceleration waves respectively, the force output of the string model is given by

$$\begin{aligned} F(z) &= Z(z)I(z)(A_1(z) - A_2(z)) \\ &= Z(z)I(z)(1 - R_b(z))A_1(z) \end{aligned} \tag{2.8}$$

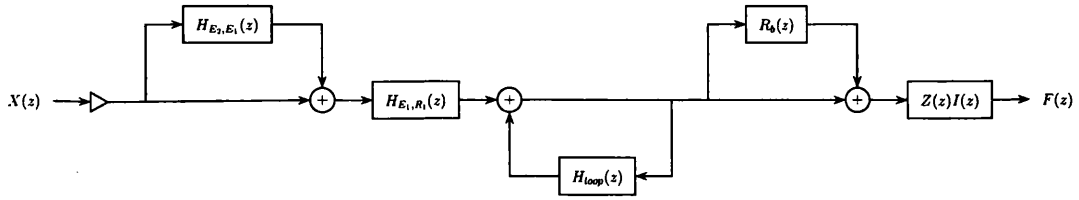

Figure 2.5: SDL model with force output at the bridge

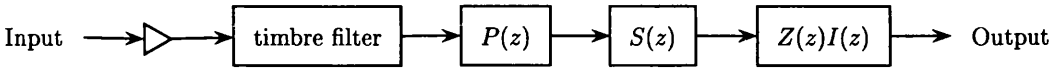

Figure 2.6: A string model with plucking-point equalizer

The acceleration signal $A_1(z)$ is evaluated as

$$\begin{aligned} A_1(z) &= A_1(z)R_b(z)H_{R_2,E_2}(z)H_{E_2,E_1}(z)H_{E_1,R_1}(z) \\ &+ X_{eq,E_1}(z)H_{E_1,R_1}(z) \\ &= H_{E_1,R_1}(z)X_{eq,E_1}(z)S(z) \end{aligned} \tag{2.9}$$

where

$S(z) = (1/1 - H_{loop}(z))$ – is the string transfer function, and

$H_{loop}(z) = R_b(z)H_{R_2,E_2}(z)H_{E_2,E_1}(z)H_{E_1,R_1}(z)$

Hence the overall transfer function of the model from excitation to bridge equals

$$H_{E,B}(z) = \frac{1}{2}(1 + H_{E_2,E_1(z)})H_{E_1,R_1}(z)S(z)(1 - R_b(z))I(z)Z(z) \tag{2.10}$$

The block diagram of the model described by the previous equation is shown in figure (2.5). For practical sound synthesis the model in figure (2.5) can be further simplified to the following model in figure (2.6). In this figure the timbre control filter is a first or second-order recursive filter that can be adjusted for softer or sharper attack [18]. Filter $P(z)$ is a plucking point equalizer that brings about the effect of the plucking point. The string model can be implemented efficiently by using a delay line and a frequency-dependent filter as shown in figure (2.7).

Note that all digital filters used in the digital waveguide model must have gain less than or equal to unity for the whole model to be stable.

2.4 The Karplus-Strong Algorithm and its Extensions

The Karplus-Strong algorithm (KS) for a plucked string was developed by Karplus and Strong in 1981 [25]. It is based on the idea that a wave-table is modified while being read. The advantage of this technique is its simplicity to implement. In addition the produced sound is rich and natural. This section describes the algorithm and its extensions [25], [18].

2.4.1 The Plucked-string Algorithm

The KS-algorithm for a plucked string is based on re-circulating a signal around a delay line through a low-pass filter called the "loop filter". The basic algorithm involves a delay line that is initially filled with random numbers and a low-pass filter. The low-pass filter commonly used in this algorithm is the two-point average filter. This filter determines its output by taking the average of the sample value currently emerging from the delay line and the value of the previous output of the delay line.

The difference equation of the string simulator is given by

$$y(n) = x(n) + \frac{y(n-L) + y(n-L-1)}{2} \tag{2.11}$$

where $x(n)$ is the input signal at sample n, $y(n)$ is the output signal at sample n, and L is the delay line length. The block diagram of the KS-algorithm is shown in figure (2.8). The transfer function of the string simulator is

$$S(z) = 1/(1 - z^{-L}H(z)) \tag{2.12}$$

where

$$H(z) = (1 + z^{-1})/2 \tag{2.13}$$

is the filter transfer function. The amplitude response is defined as the magnitude response of the frequency response of the loop filter. It gives the gain of the filter

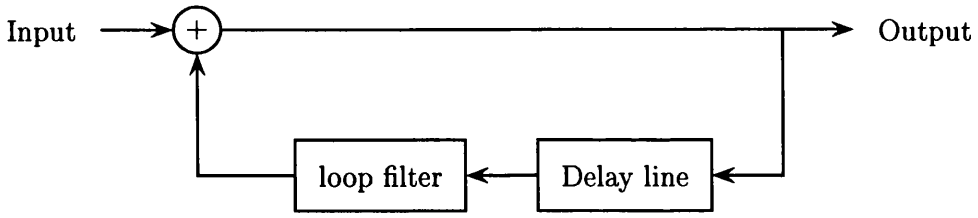

Figure 2.7: Single delay-line loop model

as a function of the frequency [48]. The amplitude response of the loop filter is given by

$$G(\omega) = |H(\omega)| = |H(z = e^{i\omega})| = \cos(\omega/2) \tag{2.14}$$

where $\omega = 2\pi f/f_s$ is the radian frequency.

Equation (2.14) shows the low-pass behaviour of the loop filter since the gain decreases with frequency according to the first quadrant of a cosine. Note that all the frequencies are restricted by the Nyquist limit. It is clear that the loop filter is necessary to bring about the decay effects of a plucked-string tone since the delay line is lossless.

The phase delay, defined as the time delay in seconds experienced by each harmonic, of the loop filter is half a sample $(T_s/2)$ sec, whereas the delay line has a phase delay equal to its length. Therefore the total loop delay is $(L + 1/2)$ samples and the period is $(L + 1/2)/f_s$ sec, which corresponds to the perceived pitch.

The KS-algorithm is just an abstract algorithm that has nothing to do with the physics of plucked strings. In 1992, J.O.Smith has interpreted the KS-algorithm physically using the digital waveguide model [49]. It is clear that the algorithm shown in figure (2.8) is comparable with the single-delay line model shown in figure (2.7). However the KS-algorithm is efficient in simulating not only plucked-string tones but also drum-like tones by changing the initial condition [25]. In

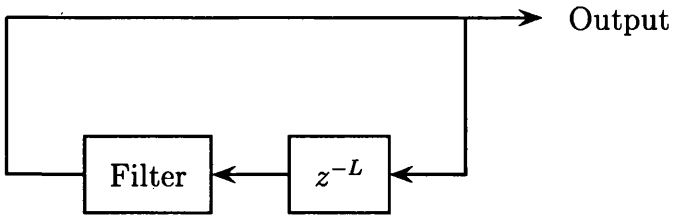

Figure 2.8: The basic KS-algorithm

addition, the algorithm is simple and easy to implement. On the other hand, there are some drawbacks of the algorithm that will be addressed in the next subsection.

2.4.2 Extensions of the KS-Algorithm

As previously mentioned the loop filter is the main lossy part of the basic KS-algorithm. According to its magnitude response, the two-point average filter is just not enough to represent the frequency-dependent damping of a real string. Different choices have been made for the loop filter by Valimaki et al., [59], [62]. Throughout this thesis, a first order IIR-filter is used as the loop filter. The difference equation of this filter is given by

$$y(n) = g(1+a)x(n) - ay(n-1) \tag{2.15}$$

where a is the filter parameter that determines its cut-off frequency and g is the filter gain at zero frequency, i.e. the DC-gain. For the filter to be stable, g must be less than unity at all frequencies and $-1 < a < 0$. The transfer function of the one-pole filter is

$$H(z) = \frac{g(1+a)}{1+az^{-1}} \tag{2.16}$$

Both the magnitude response $G_l(\omega)$ and the phase response $\theta_l(\omega)$ of this filter are given by the following equations

$$G_l(\omega) = \frac{g(1+a)}{\sqrt{1+2a\cos(\omega)+a^2}} \tag{2.17}$$

$$\theta_l(\omega) = \tan^{-1}\left(\frac{a\,\sin(\omega)}{1+a\cos(\omega)}\right) \tag{2.18}$$

The one-pole filter is still simple and easy to implement. It is also found to be sufficient for high-quality analysis [55].

Another major drawback of the basic KS-algorithm is tuning. This problem results from the fact that delay line length must be of integer value. Hence, only pitches with discrete values can be produced using this algorithm. This introduces an increasing error with frequency. A solution to this problem is by introducing a fractional-delay filter in the feed-back loop of the KS-algorithm, which must have unity gain at all frequencies [60]. A first order all-pass filter is used as the fractional delay filter with the following difference equation

$$y(n) = c\,x(n) + x(n-1) - c\,y(n-1) \tag{2.19}$$

where c is the only coefficient to be set that must be less than unity for stability. The transfer function of such a filter is

$$F(z) = \frac{c+z^{-1}}{1+c\,z^{-1}} \tag{2.20}$$

with phase response given by

$$\theta_f(\omega) = \tan^{-1}\left(\frac{-\sin(\omega)}{c+\cos(\omega)}\right) - \tan^{-1}\left(\frac{-c\sin(\omega)}{1+c\cos(\omega)}\right) \tag{2.21}$$

With these extensions the extended KS-algorithm is shown in figure(2.9). The overall transfer function of the string simulator is

$$S(z) = \frac{1}{1-z^{-L}H(z)F(z)} \tag{2.22}$$

which is fully determined by the string length and the loop filter parameters a and g as well as the fractional delay filter parameter c. The relation between the filter parameter c and the fractional delay is explained in chapter 5.

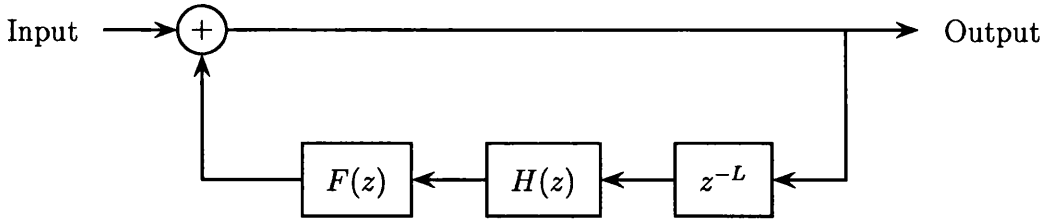

Figure 2.9: Extended KS-algorithm

2.5 The Commuted Waveguide and Body Factoring

A complete model of the acoustic guitar is divided functionally into three sub-models. These models are namely: the excitation model, the string model, and the body model. Each of these models makes its own contribution to the radiated sound. For instance, the excitation model has information about the plucking event (plucking type and plucking point) and the initial amplitude level of the radiated sound. The string model simulates the vibratory motion of the guitar string. It also brings about the decay and dispersion effects of a real guitar tone. The guitar body is considered as the resonating system [9]. It is necessary for radiating sound since strings do not radiate sound effectively due to the fact that they act as dipoles [33]. The body is also necessary for sound colouration and directivity to the produced sound. This section describes the commuted waveguide principle as well as the body-model factoring technique.

2.5.1 The Commuted Waveguide Principle

For high-quality synthesis, the acoustic guitar is modelled according to its functional structure, i.e. a complete model includes Excitation model $E(z)$, String model $S(z)$ and Body model $B(z)$. A block diagram of such a model is shown in figure(2.10) where $\delta(n)$ is the unit impulse function.

Thus the transfer function of the output of such a model is

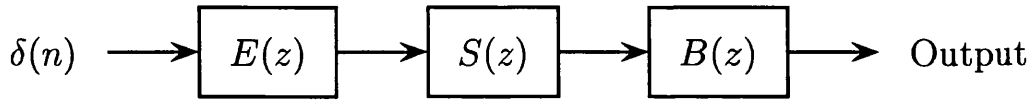

Figure 2.10: A complete model of the guitar

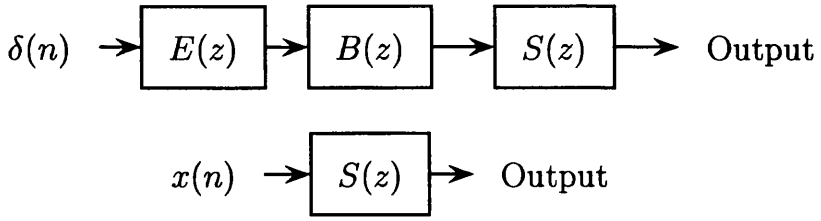

Figure 2.11: Principle of commuted waveguide

$$Y(z) = E(z)S(z)B(z) \tag{2.23}$$

The excitation is simply modelled as a wave-table or as a set of wave-tables [32]. The string model is efficiently modelled using the principle of the waveguide and the extensions of the Karplus-Strong algorithm as described in the aforementioned sections. Modelling the guitar body as a separate digital filter has been found far too expensive. This is due to the fact that the filter order must be rather high to fully model the guitar body [59]. To overcome this problem, the principle of commuted waveguide is utilized. The idea of this principle is based on commutation in linear systems.

Assuming that all three models are linear and time-invariant, the body model is commuted with the string model. Hence it is convolved with the excitation signal to yield an aggregate input signal $x(n)$ [50], [59], as shown in figure (2.11).

The advantage of commuted waveguide is that it avoids the implementation of a separate high-order body filter. Also, the aggregated input signal can be pre-computed and stored in a wave-table. On the other hand, the commuted waveguide principle has some disadvantages. For instance, the loss of parametric

control over the body modes as well as increasing memory requirement cost.

2.5.2 Body-model Factoring

Body factoring is a compromise between commuted waveguide synthesis and body modelling as a separate digital filter. In this technique the guitar model is factored into its most-damped and least-damped resonances. The least-damped resonances have the main effect on the produced sound as well as the longest decays. Thereby these resonances can be reproduced in the final sound using separate reasonable-order digital filters, while the most-damped ones are left in the input signal $x(n)$.

Karjalanien et al. have discussed methods for extracting such least-damped resonance [23]. The lowest damped resonances are the main air resonance (Helmholtz resonance), which occurs at about 100 Hz and the first mode of the top plate of the soundboard at about 200 Hz. The extracted resonances are reproduced in the synthetic sound using a second-order IIR filter. A general second-order IIR filter is given by [36]:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}} \tag{2.24}$$

where b_i, a_i, $i = 0, 1, 2$ are the filter coefficients.

Body resonators can be implemented either in parallel or in cascade with the string model where the resonator requirements are different in each case. In parallel implementation, the output of each resonator is added to the output of the string model. This implies that the magnitude response of each resonator should have a resonance peak in a vicinity determined by bandwidth and center frequency of each resonance, i.e. the resonator must attenuate all other frequencies with emphasis only on the centre frequency.

In a cascade implementation, the output of the string model is fed into the body model. In this case the body resonator must have one peak at the centre frequency and unity gain at all other frequencies. A hybrid implementation

is a compromise between two such implementations where body resonators are connected in cascade and in parallel with the string model. A more detailed discussion between the advantages and disadvantages of each implementation is provided by Tolonen [55].

A parallel implementation is chosen for this work. Note that a separate excitation signal is required for the body resonator in this implementation. Based on bilinear transformation, a second-order IIR-filter suitable for parallel implementation is given by [15]:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{2.25}$$

where:

$b_0 = 1 - b$;

$b_1 = 0$;

$b_2 = b - 1$;

$a_1 = -2b\cos(\omega_0)$;

$a_2 = 2b - 1$;

$\omega_0 = \frac{2\pi f_0}{f_s}$;

$\delta\omega = \frac{2\pi\delta f}{f_s}$ and

$b = \frac{1}{(1+\tan(\delta\omega/2))}$.

Where f_0 is the centre frequency of the resonator and δf is the resonator bandwidth in Hertz. Therefore, a body resonator is completely determined by its centre frequency and its 3-dB bandwidth. The advantages of the separate body resonator for least-damped resonances include the fact that the excitation signal is shortened due to removing the longest decay body resonance and hence less memory is required. In addition the least-damped resonance are in parametric

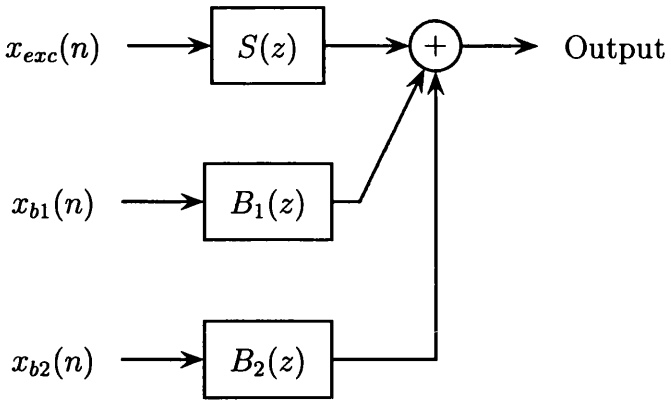

Figure 2.12: The final guitar model

form for independent control. Furthermore such body resonators can be shared by all string models, which reduces the computational burden [23].

2.6 The Guitar Model and its Extensions

The guitar model considered in this current work is basically a string model $S(z)$ together with two body resonators $B_1(z)$, $B_2(z)$ that correspond to the two least damped body resonances. The resonators are connected in parallel with the string model where each resonator has a transfer function given by equation (2.25). The guitar model is shown in figure (2.12) where $x_{exc}(n)$, $x_{b1}(n)$ and $x_{b2}(n)$ are the excitation signals for the string model and the body resonators respectively.

Such a model can be extended in several ways to better simulate a physical string. In this section three basic extensions to the guitar model are addressed. These extensions are the plucking position, sympathetic vibration, and finally the two polarizations of the vibratory motion of the string.

2.6.1 The Plucking Position

The acoustic guitar is played by plucking the string between the bridge and the body fret using a plectrum, fingertips or finger-pick. The plucking event itself

is a very complex one. It includes the plucking style, plucking position, and the interaction between the string and the player. When the string is plucked at any node of one of its vibrational modes, such a mode will not vibrate [9]. Introducing zeros uniformly distributed in the spectrum of the input signal can simulate the plucking position effect. In other words, this is done by filtering the input signal by a comb filter [18]. However, estimating the plucking point by searching for the missing modes in the spectrum is a difficult and unsolved task [55], [58]. Modelling the plucking position is not considered in the current work and its contribution is left in the excitation signal as will be described in chapter 5.

2.6.2 Sympathetic Vibration

The phenomena of sympathetic vibration occur in stringed musical instruments when a string is set into motion by the vibration of another string. In this case all the partials of the plucked string that do not coincide with those of the sympathetic string will be highly attenuated. Meanwhile, the partials of the plucked string that coincide with those of the sympathetic vibration will strongly resonate. The effect of sympathetic vibration is incorporated in the guitar model by feeding in a copy of the string simulator with a small percentage of the output of another (plucked) string with different string length [18], [55]. Sympathetic vibration adds a reverberant feel to the produced sound as well as giving each pitch in the range of the instrument an individual character. Nevertheless the player can control these phenomena by damping the non-plucked strings.

2.6.3 Vertical and Horizontal Vibration of the String

The vibratory motion of a string is divided into longitudinal, torsional and transversal motions. While longitudinal and torsional vibrations are not prominent in sound-synthesis of the acoustic guitar, they can be of significant importance in other stringed instruments [55], [24]. Hence the vibration of the string

is regarded as transversal vibration. In this kind of vibration, every point on the string vibrates in a plane that is normal to the string. The transversal vibration is further divided into horizontal (vibration along the soundboard) and vertical (vibration normal to the soundboard). It is also expressed as the sum of relatively related modes obtained by solving the wave equation of the string as described in section (2.2). The two polarizations vibrate in a slightly different manner which results in slightly different frequencies that create beats in the resulting sound. The two-stage decay rate of the guitar tones is another phenomenon of such dual-polarization [55]. In order to simulate this duality in the guitar model, two digital waveguides with a different L for each string are used for every vibration plane [59], [24]. In our model only one of these polarizations is considered.

2.7 Summary

In this chapter, the basic principles of *digital waveguide modelling* as an efficient physical modelling technique for digital sound synthesis are presented. This technique has been applied to a vibrating string which is the main source of vibration in any stringed musical instrument. The basic elements of such a model are delay lines and digital filters. This chapter also describes the equivalence between a bi-directional delay-line waveguide model and single-delay waveguide model. The Karplus-Strong algorithm for simulating plucked string tones and drum-like tones, as an abstract algorithm, is also presented. Some of the main extensions to the basic algorithm have been considered. Based on Karplus-Strong extensions and the principle of the commuted waveguide model, a complete model of the acoustic guitar has been designed. Further extensions and other non-linearities to the model have also been addressed in this chapter.

Chapter 3

A Wavelet-based Pitch Detector

3.1 Introduction

In the previous chapter a physical model of the acoustic guitar is designed based on the principle of commuted digital waveguide and the extensions of the KS-algorithm. Once the model is designed, to simulate closely the physical behaviour of a real guitar and to be efficiently realizable in real time, its parameters have to be extracted. Thus the objective of the analysis process is to derive the string model parameters $S(z)$, the body resonators' $B_1(z)$, $B_2(z)$ parameters and finally the excitation signals $x_{exc}(n)$, $x_{b1}(n)$, and $x_{b2}(n)$.

A general overview of the analysis process is shown in figure (3.1) which starts by estimating the pitch period of a recorded real guitar tone. This step provides the total delay of the string model since

$$L_1 = \frac{f_s}{f_0} \tag{3.1}$$

where f_0 is the estimated pitch period.

Unlike speech signals musical signals have a broad range of frequencies so there are some difficulties in estimating their pitch period. The autocorrelation function is one of the well-known time-domain pitch detectors. Despite its simplicity, the autocorrelation function has some disadvantages as will be explained further in the following section.

In this chapter an algorithm based on the Dyadic Wavelet Transform (DWT) is proposed for pitch estimation of musical signals. For this purpose a quadratic spline wavelet with certain criteria is constructed. The performance of both linear phase wavelets and minimum phase wavelets is also investigated. Furthermore a comparative study between this algorithm and the autocorrelation function is given. The new algorithm is applied not only to different guitar signals but also to a wide range of musical signals as well as some singing voices. The proposed algorithm is proven to be simple, accurate, fast, robust to noise and has the potential for real-time applications.

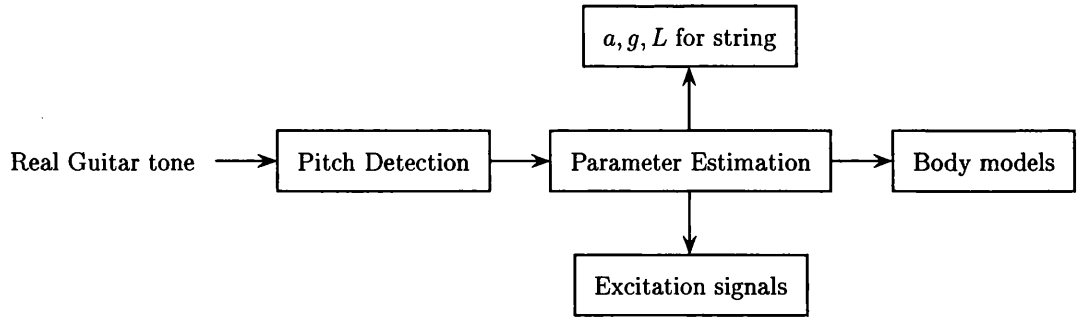

Figure 3.1: An overview of the analysis process

This chapter is organized as follows: section 2 is devoted to the pitch detection problem and the autocorrelation algorithm. The performance of various window functions is evaluated. In section 3 principles of the Dyadic Wavelet Transform and its properties are presented. The construction of the wavelet used in the analysis is given in section 4 along with some other wavelets. In section 5 the implementation of the proposed algorithm is described. The algorithm is also tested for different musical signals and some singing voices. Results and discussion are presented in the last section.

It is worth mentioning here that the whole analysis process will be carried out on a single guitar tone to extract all the model parameters. In appendix B the analysis process will be examined for different guitar signals.

3.2 Pitch Detection and the Autocorrelation Function

The pitch period is a fundamental parameter in the analysis process of any physical model. A pitch detector is basically an algorithm that determines the fundamental pitch period of an input musical signal. Pitch detection algorithms can be divided into two groups: time-domain pitch detectors and frequency-domain pitch detectors. A time-domain pitch detector is applied directly to the input signal. It estimates the pitch period by trying to find repeating patterns in

the input signal. Zero-crossings, the autocorrelation function and the adaptive filtering are among the various techniques of time-domain pitch detectors. A frequency-domain pitch detector applies a Fourier Transform on a short segment of the signal. If the signal is periodic, then the spectrum has peaks at multiples of the fundamental. Examples of this category are the cepstrum and the maximum likelihood methods. In general, estimating the pitch period of a musical signal is not a trivial task due to some difficulties such as the attack transients, low frequencies, and high frequencies. For a detailed discussion about the various pitch detection techniques, see Roads [42].

In this current work, we consider the autocorrelation function as a well-known, time-domain pitch detector. It is a measure of similarity between a signal and translated (shifted) version of itself. The basic idea of the autocorrelation function is that periodicity of the input signal implies periodicity of this function and vice versa. For non-stationary signals the short-time autocorrelation function is defined as [39]

$$Ph_l(m) = \frac{1}{N} \sum_{n=0}^{N-m-1} [x(n+l)w(n+l)][x(n+m+l)w(n+m+l)] \qquad (3.2)$$

for $0 \leq m \leq M_0 - 1$.

where $w(n)$ is an appropriate window function, N is the frame size, l is the starting frame index, m is the autocorrelation parameter or time lag and M_0 is the total number of points to be computed in the autocorrelation function. In general the window function is an even function of compact support [38]. The effect of the window function is to taper the input signal smoothly to zero. Therefore the frame size N has to be at least four times the fundamental period and the input signal $x(n)$ is assumed to be stationary within each frame.

The autocorrelation function has its highest peak at $m = 0$ which equals to the average power of the input signal. To estimate the pitch period one searches for the local maxima in a meaningful range of m for each frame l. The distance

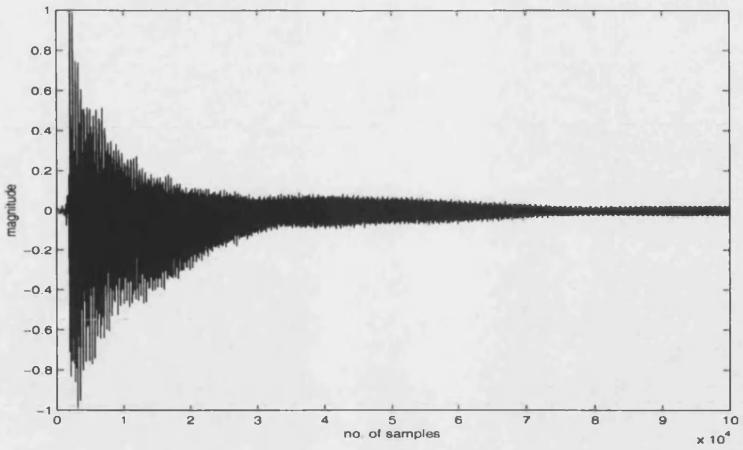

Figure 3.2: The guitar test signal

between two consecutive maxima is the pitch period of the input signal $x(n)$. The estimated values of the local maxima can be further improved by using parabolic interpolation which is presented in more details in appendix A. Different window functions such as the rectangular, Hanning, Hamming, and Blackman windows are used in the analysis [38].

Table (3.1) shows the estimated pitch period for a particular recorded real guitar signal using various windows. The numbers given result from an implementation of the algorithm, but the results beyond the first decimal place can not have any reliance. Figure (3.2) shows the underlying guitar test signal which is an E2-tone with pitch period 330.4001Hz. The analysis starts after 226.75ms of the signal to avoid the attack transient. The window size is 22.7ms for all window functions and the sampling frequency f_s is 44100 Hz.

Table (3.2) shows the estimated pitch period of the guitar tone using the Hamming window but with different frame size. It is clear that the estimated pitch differs slightly with the chosen frame size as well as the analysis window.

Despite the simplicity of the autocorrelation function it has some limitations. The choice of the window function, the frame size and the assumption that the

Window function	Estimated pitch(Hz)
Rectangular	330.3488
Hanning	330.4684
Hamming	330.4246
Blackman	330.5132

Table 3.1: Estimated pitch period using different windows

Window size(ms)	Estimated pitch(Hz)
N=12.15	330.8757
N=18.23	330.6094
N=22.67	330.4246

Table 3.2: Estimated pitch using various frame size of the Hamming window

signal is stationary within each frame are among the main disadvantages of the autocorrelation function.

3.3 The Wavelet Transform

This section serves as a basic introduction to the wavelet transform. The reader familiar with the wavelet theory can skip this section to the next one. The section starts by describing the fundamental requirements for a window function to be a wavelet. The **Continuous Wavelet Transform** (CWT) as a time-frequency analysis tool is described next along with its properties. This section continues by describing a discrete version of this transform which is the **Dyadic Wavelet Transform** (DWT). Finally properties of the DWT are presented with emphasis on the one that is proven to be useful in pitch detection.

3.3.1 The Continuous Wavelet Transform

The Continuous Wavelet Transform is a tool for time-frequency localization of non-stationary signals. It is based on a window function $\psi(t)$, called the *mother wavelet* or the *analysis wavelet*, which has to satisfy some requirements [26]:

(i) The mother wavelet has to be of finite energy. That is

$$\psi(t) \in L^2(R)$$

where $L^2(R)$ is the space of all real-valued functions of finite energy.

(ii) $\psi(t)$ satisfies the admissibility condition which also implies that its mean value should be zero.

$$C_g = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty \tag{3.3}$$

where $\hat{\psi}(\omega)$ is the Fourier transform of $\psi(t)$, i.e. $\int_{-\infty}^{\infty} \psi(t)dt = 0$.

In other words the mother wavelet has to oscillate (i.e changes sign) as a wave in R. The admissibility condition is a necessary condition for the inversion of the CWT.

(iii) $\psi(t)$ has to decay fast, i.e decreases as $t \to -\infty$ or ∞. A wavelet with compact support trivially satisfies this condition.

For a given mother wavelet $\psi(t)$, the Continuous Wavelet Transform is defined as [43]:

$$W_{a,b}(f)(t) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \overline{\psi(\frac{t-b}{a})} dt \tag{3.4}$$

where $a \in R^+$ is called the dilation (scale) parameter, $b \in R$ is the position parameter that indicates where the mother wavelet is located, and $\overline{\psi(t)}$ denotes the complex conjugate of $\psi(t)$. Equation (3.4) measures the similarities between a time-domain signal $f(t)$ and the wavelets $\psi_{a,b}(t)$ where

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi(\frac{t-b}{a})$$

The coefficient $\frac{1}{\sqrt{a}}$ is for energy conservation. Viewing the wavelet as a window function, we find that for large a (low frequency) the width widens whereas

it is narrowing for small a (high frequency). Equation (3.4) is a time-domain representation of the continuous wavelet transform which can be viewed as a convolution (filtering) of the signal $f(t)$ with a dilated version of the mother wavelet. Its frequency-domain representation is given by

$$W_{a,b} = \sqrt{a} \int \int_{a,b} F(\omega)\, \overline{\psi(a\omega)} d\omega \tag{3.5}$$

where $F(\omega)$ is the Fourier Transform of $f(t)$ which can be recovered from its CWT by using

$$f(t) = \frac{1}{C_g} \int \int_{a,b} W_{a,b}\, \psi_{a,b}(t)\, \frac{dadb}{a^2} \tag{3.6}$$

where C_g is as defined in equation (3.3). Equation (3.6) is called the *synthesis equation.*

3.3.2 Properties of the Continuous Wavelet Transform

In addition to its simple interpretation the CWT exhibits some other useful properties such as:

- CWT is a linear transformation.
- It satisfies the *covariance property* with respect to dilation and translation of the signal $f(t)$. For a fixed mother wavelet $\psi(t)$, if $W_{a,b}$ is the continuous wavelet transform of a given signal $f(t)$, i.e. $f(t) \rightarrow W_{a,b}$ then

$$\begin{aligned} f(t - t_0) &\rightarrow W_{a,b-t_0} \\ \frac{1}{\lambda} f(t/\lambda) &\rightarrow W_{a/\lambda, b/\lambda} \end{aligned}$$

  for $\lambda > 0$.
- Conservation of energy. The Parseval's identity for the Continuous Wavelet Transform is given by

$$\int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{C_g} \int \int_{a,b} |W_{a,b}|^2\, \frac{dadb}{a^2} \tag{3.7}$$

That is, the energy of signal $f(t)$ can be obtained, up to a constant, from the energy of its CWT. In other words, the CWT can be considered as an isometric operator.

- Zooming-in property which is the most significant property of the CWT. In Short Time Fourier Transform (STFT) the width of the analysis window is fixed over the entire time-frequency plane. Contrary to the STFT, the CWT zooms in on the fine details (large scale) of the signal and zooms out for coarser trends of the signal (small scale). More about the Short-Time Fourier Transform is considered in the next chapter.

- Constant-Q analysis. The CWT satisfies the additional property of *constant-Q* analysis. That is, the ratio between frequency resolution and frequency is always constant. This frequency-dependent resolution is the basis for the application of wavelet analysis to multi-resolution analysis and sub-band coding [54] .

3.3.3 The Dyadic Wavelet Transform

In the CWT both the scale parameter a and the position parameter b are continuous variables. Therefore the CWT is computationally very intensive and has a significant amount of redundant information. To reduce the computational complexity, both parameters a and b are made discrete. In the discrete case only positive values of the scale parameter a will be taken into account. One of the possible different ways to discretise the parameters is

$$a = a_0^j, \qquad b = k\, b_0\, a_0^j$$

where j and k are integers, $a_0 > 0$ and $b_0 > 0$ are fixed. For all practical purposes a_0 will be set equal to 2 and b_0 will be set equal to 1. That is $a = 2^j$ and $b = k2^j$. This sampling of a and b is called the *critical sampling.*

The Dyadic Wavelet Transform (DWT) of an arbitrary signal $f(t) \in L^2(R)$ is defined as [27]

$$DWT_j(f) = f(t) * \psi_j(t) \tag{3.8}$$

where $*$ denotes convolution, and

$$\psi_j(t) = \frac{1}{2^j}\psi(\frac{1}{2^j})$$

In addition to its linearity and shift-invariance, DWT is sensitive to points of sharp variations or discontinuities of the signal, exhibiting local maxima around the points of sharp variation [27], [28], and [3]. This property is proven to be very useful in image processing, edge detection and pitch detection of speech signals [21].

An appropriately chosen wavelet for the purpose of pitch detection is a wavelet that is the first derivative of a smooth function [6]. Zero-crossings of musical signals can be considered as points of sharp variation and hence the DWT exhibits local maxima at these points across several consecutive scales. The pitch period is evaluated by measuring the time distance between two such consecutive maxima.

3.4 Construction of the Analysis Wavelet

In general a basic wavelet has to satisfy a minimum set of constraints as mentioned before. For certain applications further conditions could be imposed on the analysis wavelet such as orthogonality, compactness or linear phase. In this section we construct the quadratic spline wavelet as the main analysis wavelet. Some other wavelets are described in this section as well. We start by presenting the basic mathematics behind the multi-resolution analysis and the construction of a minimum-phase wavelet. Such mathematical background is required for the construction of the quadratic spline wavelet. Finally this section presents another example of a linear phase wavelet which is the Morlet wavelet.

3.4.1 Wavelets and Multi-resolution Analysis

The construction of an orthonormal family of basis function for $L^2(R)$ starts with the solution $\phi(t)$ of a dilation equation (two-scale equation). The wavelet $\psi(t)$ is then obtained from $\phi(t)$, which is called the *scale function.* The basic form of a dilation equation is given by [53]

$$\phi(t) = \sum_{k=0}^{N-1} C_k \, \phi(2t - k) \tag{3.9}$$

where N is the number of the wavelet coefficients, and $C_k \in l^1$ are the wavelet coefficients or the refinement coefficients. Since there are only N nonzero coefficients in (3.9), $\phi(t)$ has a compact support $[0, N-1]$. It is difficult to obtain a solution $\phi(t)$ by directly solving equation (3.9) due to the presence of the scale factor 2 [54]. Nevertheless a solution $\phi(t) \in l^1$ is ensured such that

$$\int_{-\infty}^{\infty} \phi(t)dt = 1$$

by imposing

$$\sum_{k=0}^{N-1} C_k = 2 \tag{3.10}$$

This condition is called the *conservation of area* condition. It is worth mentioning here that condition (3.10) ensures the uniqueness of $\phi(t)$ but not the smoothness [53]. There are three different methods for solving equation (3.9):

1. The iteration method,
2. The recursive method, which is the most efficient way to evaluate $\phi(t)$ and
3. The Fourier analysis method. This method gives the scale function $\phi(t)$ as a distribution rather than an exact formula.

All the wavelets which have been used in this study are constructed using the recursive method as will be further explained in the following subsection.

Taking the Fourier transform of equation (3.9) yields

$$\hat{\phi}(\omega) = H(\frac{\omega}{2})\,\hat{\phi}(\frac{\omega}{2}) \tag{3.11}$$

with

$$\begin{aligned} H(\omega) &= \frac{1}{2}\sum_{k} C_k\, e^{-i\omega k} \\ &= \sum_{k} h_k\, e^{-i\omega k} \end{aligned} \tag{3.12}$$

where $H(\omega)$ is the Fourier transform of a set of non-zero coefficients $\{h_k\} = \{C_k/2\}$. It is a continuous and 2π-periodic function with $H(0) = 1$. For a family of the scale function $\phi(t)$ and its translations to form an orthonormal family, i.e

$$\int_{-\infty}^{\infty} \phi(t)\,\phi(t-m)dt = 0, \quad for \quad m \neq 0$$

and

$$\int_{-\infty}^{\infty} |\phi(t)|^2\, dt = 1 \tag{3.13}$$

the wavelet coefficients have to satisfy

$$\sum_{k=0}^{N-1} C_k\, C_{k+2m} = 0 \tag{3.14}$$

for all $m \neq 0$, with the additional condition

$$\sum_{k=0}^{N-1} C_k^2 = 2 \tag{3.15}$$

Conditions (3.14) and (3.15) are known as the orthonormality conditions. Equivalently the *orthonormality conditions* in the frequency domain are given by

$$|H(\omega)|^2 + |H(\omega+\pi)|^2 = 1$$

For approximation with accuracy p, the Fourier transform of the scale function $\phi(t)$ must have zero of order p at $\omega = 2\pi k$, $k \neq 0$. This condition turns out to be a condition imposed on the wavelet coefficients such that

$$\sum_{k=0}^{N-1} (-1)^k\, k^m\, h(k) = 0 \tag{3.16}$$

for $m = 0, 1, \ldots, p-1$. Equivalently in the frequency domain, $H(\omega)$ must have zero of order p at $\omega = \pi$. That is

$$\frac{d^k}{d\omega^k} H(\omega)|_{(\omega=\pi)} = 0$$

for $k = 0, 1, \ldots, p-1$. This condition is called the *accuracy condition.* It implies that $H(\omega) = 0$ at $\omega = \pi$. In other words

$$\sum_{k\,even} h(k) = \sum_{k\,odd} h(k) = 1/2 \tag{3.17}$$

From multi-resolution analysis, the wavelet function $\psi(t)$ is constructed from its scale function $\phi(t)$ by the equation

$$\begin{aligned} \psi(t) &= \sum_{k=0}^{N-1} d_k\, \phi(2t-k) \\ &= \sum_{k=0}^{N-1} (-1)^k\, C_{N-k-1}\, \phi(2t-k) \end{aligned} \tag{3.18}$$

That is $\psi(t)$ has the same coefficients as the scale function $\phi(t)$ but in reverse order with alternating sign. This construction implies that $\psi(t)$ has a compact support $[0, N-1]$ and orthogonal to its scale function $\phi(t)$ and its translations $\phi(t-m)$, i.e.

$$\int_{-\infty}^{\infty} \psi(t)\phi(t-m)\, dt = 0$$

for all m. The Fourier transform of $\psi(t)$ is

$$\hat{\psi}(\omega) = G(\frac{\omega}{2})\hat{\phi}(\frac{\omega}{2}) \tag{3.19}$$

with

$$G(\omega) = \frac{1}{2}\sum_k d_k\, e^{-i\omega k} \tag{3.20}$$

$G(\omega)$ is the Fourier transform of a finite family of nonzero coefficients $\{g_k\} = \{d_k/2\}$. Again it is a continuous and 2π-periodic function with $G(0) = 0$. It

follows from condition (3.18) that the mother wavelet is orthogonal to its own translations with or without dilation. Therefore

$$\int_{-\infty}^{\infty} \psi(t)\,\psi(t-m)dt = 0$$

for all m such that $m \neq 0$, and

$$\int_{-\infty}^{\infty} \psi(t)\,\psi(2t-m)dt = 0$$

The conclusion is that the family $\{\phi(t), \psi_{j,k},\ j \geq 0, k \in Z\}$ forms an orthonormal basis for the space $L^2(R)$ and hence, any function $f(t) \in L^2(R)$ can be expressed as a linear combination of members of such a family. This family is called Daubechies' wavelets [8]. Daubechies' wavelets are continuous, of compact support and of certain regularity. Except for the trivial case of the Haar wavelet these wavelets are neither symmetric nor anti-symmetric wavelets [7]. Two examples of Daubechies' wavelets are described next.

3.4.1.1 The Haar Wavelet

The Haar wavelet is the simplest form of a wavelet and probably the oldest. It has been known in the literature since 1910 [13]. It is defined as

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1/2 \\ -1 & \text{if } 1/2 \leq t < 1 \\ 0 & \text{otherwise} \end{cases}$$

and its corresponding scale function is given by

$$\phi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1 \\ 0 & \text{otherwise} \end{cases}$$

The Haar wavelet has only two nonzero coefficients $C_0 = C_1 = 1$. The Haar basis is an orthonormal family of compactly supported wavelets. It is a special case of Daubechies' wavelets for $N = 2$. The Haar wavelet is the only orthonormal

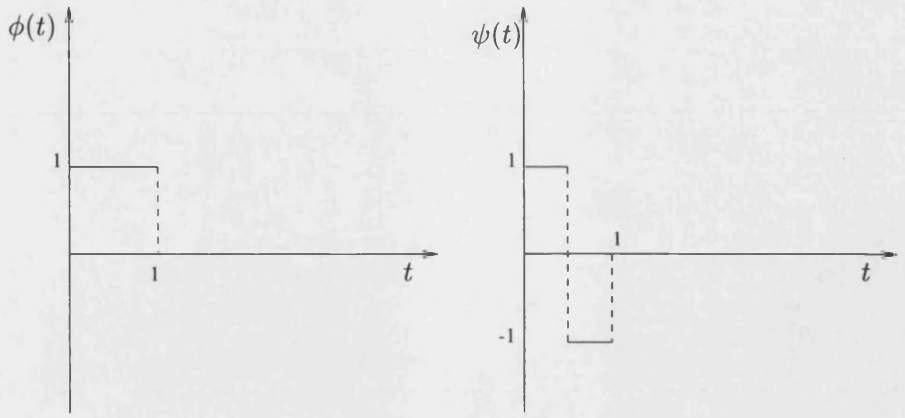

Figure 3.3: Haar scale function and its corresponding wavelet

wavelet with linear phase since it has a symmetry axis. The disadvantage of the Haar wavelet is that it is a discontinuous function. Figure (3.3) shows the Haar scale function and its corresponding wavelet.

3.4.1.2 The D4-wavelet

The D4-wavelet is a continuous wavelet that has only four nonzero coefficients in its dilation equation, hence it has a compact support [0,3]. This wavelet does not have a closed algebraic form. Nevertheless it can be constructed numerically using the previous multi-resolution technique. The wavelet coefficients are

$$C_0 = \frac{1+\sqrt{3}}{4}, C_1 = \frac{3+\sqrt{3}}{4}$$
$$C_2 = \frac{3-\sqrt{3}}{4}, C_3 = \frac{1-\sqrt{3}}{4}$$

The construction of this wavelet starts by constructing its corresponding scale function which satisfies the following two conditions

$$\begin{aligned} \phi(0) = \phi(3) &= 0, \\ \phi(1) + \phi(2) &= 1 \end{aligned}$$

By knowing the values of the scale function at integer values, then its values at all dyadic points $\{x = k/2^j,\ j, k \in Z\}$ are readily known from the dilation equation.

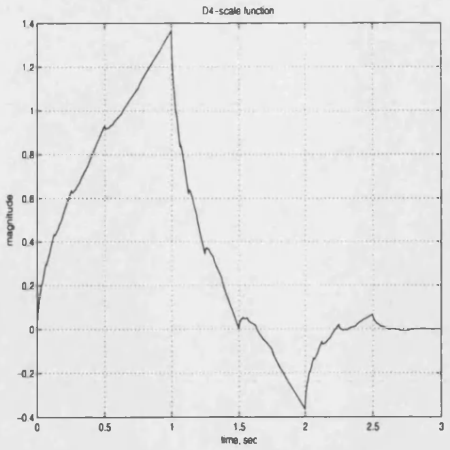

Figure 3.4: D4-scale function

This dilation equation can be considered as an eigen-value equation that takes the form

$$TM = M$$

where

$$\begin{aligned} T &= [C_{2i-j}], \qquad 1 \leq i, j \leq N-2, \\ M &= [\phi(1)\, \phi(2)]^T \end{aligned}$$

and T stands for the vector transpose. Hence M is an eigen-vector corresponding to the eigen-value $\lambda = 1$. Once all the values of the scale function $\phi(t)$ are known, the construction of the mother wavelet follows from equation (3.18). Figure (3.4) shows the D4-scale function. The corresponding wavelet is shown in figure (3.5).

3.4.2 Construction of the Quadratic Spline Wavelet

Splines are piecewise polynomials with smooth fit. They have some properties that make them very useful in practice [57]. Unlike Daubechies' wavelets, splines

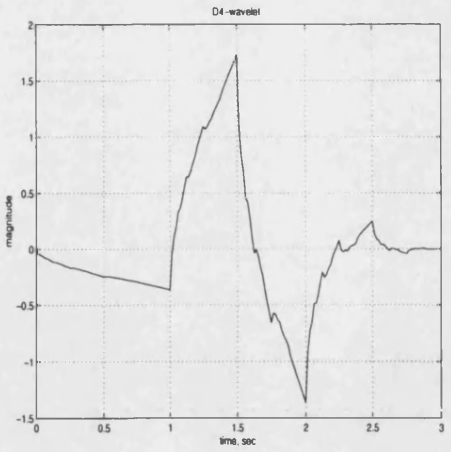

Figure 3.5: D4-Wavelet

are either orthogonal or of compact support but not both. That is, to retain orthogonality, compactness has to be dropped. In general, a spline $\phi_N(t)$ of order N has the following characteristics:

- It is a piecewise polynomial of order N with compact support $[0, N+1]$.
- It is not only continuous but also has $(N-1)$-continuous derivatives, i.e. $\phi_N(t) \in C^{N-1}$.
- It is of accuracy $p = (N+1)$, i.e. the frequency response of the corresponding low pass filter has a zero of order $(N+1)$ at $\omega = \pi$.
- It is a smooth and even function with certain regularity.
- The construction of $\phi_N(t)$ is the convolution of the unit box $\phi_0(t)$ function $(N+1)$-times. That is

$$\phi_N(t) = \underbrace{\phi_0(t) * \phi_0(t) * \ldots * \phi_0(t)}_{N+1}$$

- The refinement coefficients are the binomial coefficients.
- It has a closed form in both time and frequency domains.

The simplest example of a spline function is the Haar scale function $\phi_0(t)$ defined as before. It is a spline of order zero with accuracy $p = 1$. Its Fourier transform is given by

$$\hat{\phi}_0(\omega) = e^{(-i\omega/2)} \left(\frac{\sin(\omega/2)}{(\omega/2)} \right) \tag{3.21}$$

and the frequency response of the corresponding low pass filter is

$$\begin{aligned} H(\omega) &= \left(\frac{1 + e^{-i\omega}}{2} \right) \\ &= e^{(-i\omega/2)} \cos(\omega/2) \end{aligned} \tag{3.22}$$

In addition to the previous characteristics, the flexibility in designing corresponding wavelets with certain criteria is the main advantage of spline functions. For pitch detection purposes the analysis wavelet has to be smooth, regular, and antisymmetric with compact support and the first or second derivative of a smooth function [28]. The analysis wavelet used in this thesis for pitch detection of musical signals is a quadratic spline wavelet. The construction of such a wavelet starts with the scale function that defines it. This is the quadratic spline $\phi_2(t)$ with the Fourier transform given by

$$\begin{aligned} \hat{\phi}(\omega) &= H(\frac{\omega}{2})\, \hat{\phi}_2(\frac{\omega}{2}) \\ &= e^{(-i3\omega/2)} \left(\frac{\sin(\omega/2)}{(\omega/2)} \right)^3 \end{aligned} \tag{3.23}$$

where

$$\begin{aligned} H(\omega) &= \left(\frac{1 + e^{(-i\omega)}}{2} \right)^3 \\ &= e^{(-i3\omega/2)} \cos(\omega/2)^3 \end{aligned} \tag{3.24}$$

It is the frequency response of the low pass filter $\{h(k)\} = \{1/8, 3/8, 3/8, 1/8\}$. Note that $H(\omega)$ satisfies

$$H(0) = 1,$$

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 \leq 1$$

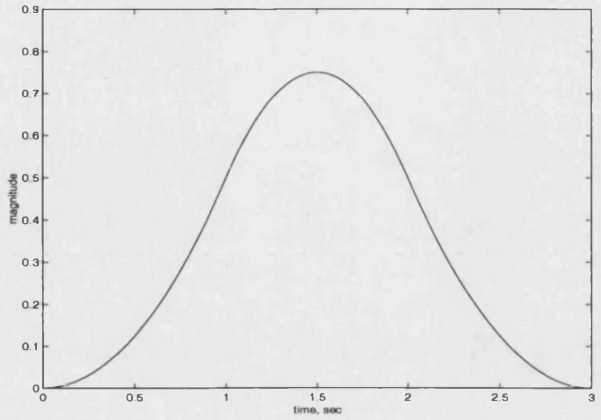

Figure 3.6: The quadratic spline scale function

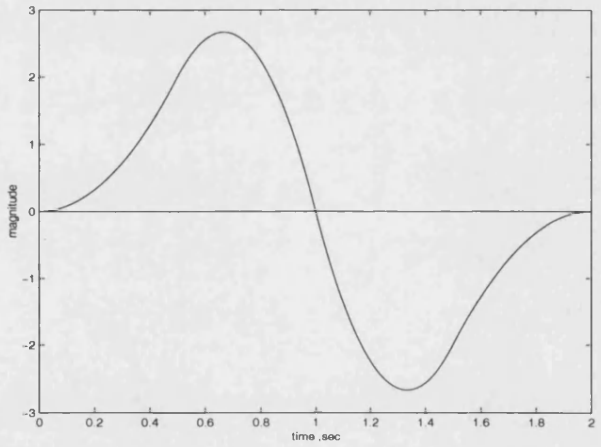

Figure 3.7: The quadratic spline wavelet

The Fourier transform of the corresponding wavelet $\hat{\psi}(\omega)$ is given by

$$\hat{\psi}(\omega) = G(\omega/2)\,\hat{\phi}(\omega/2) \tag{3.25}$$

where $G(\omega)$ is a continuous, 2π-periodic function. It is the frequency response of a high pass filter $\{g(k)\}$ such that $G(\omega) = 0$.
Substituting from equation (3.23) into equation (3.25) yields

$$\hat{\psi}(\omega) = e^{(-i3\omega/4)}\,G(\omega/2)\left(\frac{\sin(\omega/4)}{(\omega/4)}\right)^3 \tag{3.26}$$

Choosing this wavelet to be the first derivative of a cubic spline implies that it has a zero of order one at $\omega = 0$. That is

$$\begin{aligned} \psi(t) &= \frac{d\theta(t)}{dt}, \\ \hat{\psi}(\omega) &= i\omega\hat{\theta}(\omega) \end{aligned} \tag{3.27}$$

For the analysis wavelet to have the smallest support, we will consider the cubic spline given by

$$\hat{\theta}(\omega) = e^{(-i\omega)}\left(\frac{\sin(\omega/4)}{(\omega/4)}\right)^4 \tag{3.28}$$

It is clear that $\theta(t)$ is a smooth function. Hence

$$\hat{\psi}(\omega) = i\omega\, e^{(-i\omega)}\left(\frac{\sin(\omega/4)}{(\omega/4)}\right)^4 \tag{3.29}$$

Equating equation (3.26) and (3.29), yields

$$\begin{aligned} G(\omega) &= 4i\, e^{(-i\omega/2)}\sin(\omega/2) \\ &= 2(1 - e^{(-i\omega)}) \end{aligned} \tag{3.30}$$

Hence $\psi(t)$ is an anti-symmetric wavelet, regular with small finite support [0,2]. Its corresponding high pass filter is given by $\{g(k)\} = \{2, -2\}$.

Figure (3.6) shows the quadratic spline scale function $\phi(t)$. The quadratic spline wavelet is shown in figure (3.7).

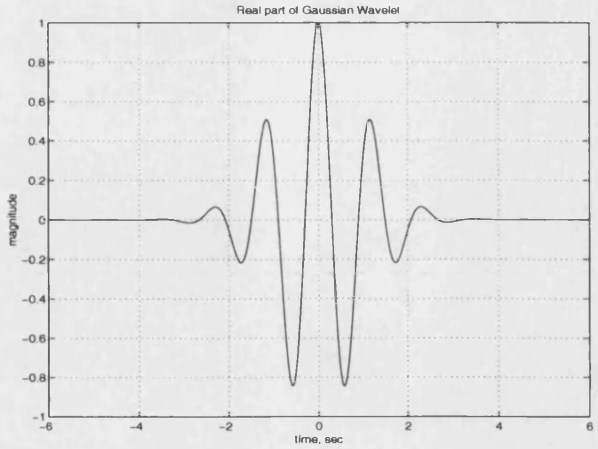

Figure 3.8: Real part of Morlet wavelet

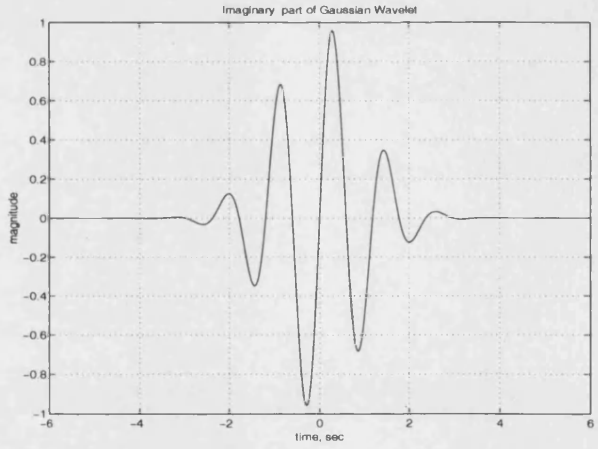

Figure 3.9: Imaginary part of Morlet wavelet

3.4.3 The Gaussian Wavelet

The Gaussian wavelet is basically a complex-sinusoid modulated with a Gaussian envelope. It is sometimes called the *Morlet wavelet* [20]. This wavelet is defined as

$$\psi(t) = e^{(i\omega_0 t)}\, e^{(-t^2/2)} \tag{3.31}$$

where its Fourier transform is defined as

$$\hat{\psi}(\omega) = e^{\frac{-(\omega-\omega_0)^2}{2}} \tag{3.32}$$

It is worth mentioning here that the Morlet wavelet is not a mother wavelet since it does not satisfy the admissibility condition. Nevertheless such a wavelet is admissible when the constant ω_0 satisfies the condition $5 < \omega_0 < 6$ [26]. The reason for including this wavelet is that its performance has been evaluated as a linear phase wavelet for pitch detection purposes. The real and imaginary parts of the Morlet wavelet for $\omega_0 = 5.3$ are shown in figures (3.8) and (3.9) respectively.

3.5 Algorithm Implementation

In this section we present the implementation aspects of the proposed algorithm. As previously mentioned the DWT has to be evaluated, theoretically, for all scales 2^j for all j varying from $-\infty$ to $+\infty$. In practice since the input signal is generally measured with a finite resolution, a finite large scale and a nonzero fine scale limit the implementation. The fine scale is set equal to 1 (for normalization purposes) and the large scale is set equal to 2^J [3]. Two FIR filters, namely, h and g characterize the DWT as well as the number of levels J. Such two filters h and g are the impulse responses of a low pass filter and a high pass filter respectively.

Starting with the assumption that $S_0 f = f$, $h_0 = h$, and $g_0 = g$ where f is a

discrete input signal, the recursive algorithm is defined as

$$\begin{aligned} W_{j+1}f &= g_j * S_j f, \\ S_{j+1}f &= h_j * S_j f \end{aligned} \tag{3.33}$$

for $j = 0, 1, \ldots, J-1$ and * denotes convolution. Both h_j and g_j denote the filters obtained from h and g by inserting $2^j - 1$ zeros between each two consecutive coefficients of the two filters respectively.

At each level j the effect of the operator S is to smooth the input signal, i.e to attenuate its high frequency components without altering its low frequency components. On the other hand the operator W gives the details lost when the signal is transformed from the previous scale to this new scale. Therefore the DWT can be implemented as a FIR non-subsampled octave-band filter bank as shown in the figure (3.10). The main analysis wavelet used in the implementation is the quadratic spline wavelet with Fourier transform given by equation (3.29). The motivation behind the proposed algorithm is that points of zero-crossings of a musical signal (glottal closure of speech signals) are considered as points of sharp variations of this signal. Hence the transform exhibits local maxima at these points across several scales [28].

For each scale j, locate the positions of the local maxima of DWT that exceed a certain threshold with respect to the global maxima and within a certain localization error. The points of zero crossings of a musical signal should have local maxima at the same points across several consecutive scales. For a certain scale the time distance between two consecutive such maxima gives the pitch period of the musical signal. The estimated values can be further improved by using several methods for curve fitting for best estimate of the local maxima. The parabolic interpolation, as previously described in the autocorrelation function, has been found adequate. The sampling rate for all test signals is 44.1 kHz and different window sizes of 22.7 and 34 ms have been used during the analysis. Experiments

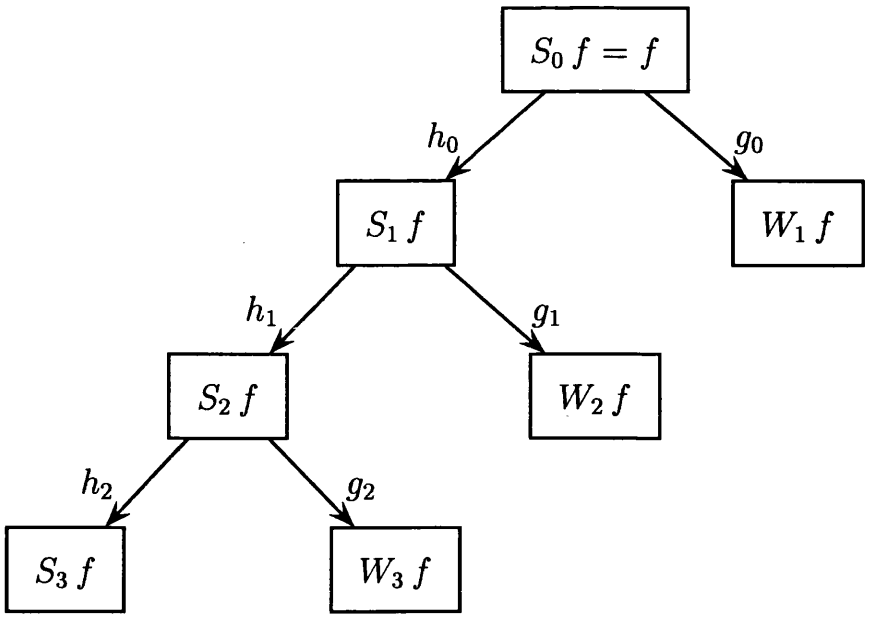

Figure 3.10: A FIR non-subsampled filter bank implementation of DWT

have shown that it is adequate to evaluate the dyadic wavelet transform across three consecutive scales only: 2^4, 2^5 and 2^6.

In addition to the quadratic spline wavelet, the performance of both minimum phase wavelets (Daubechies' wavelets) and linear phase wavelets (Haar and Morlet) is also investigated. Our main interest is the analysis of the acoustic guitar tones. However the proposed algorithm has been implemented on a wide range of other musical signals such as a saxophone signal (wind instrument), a tanpura signal (an Indian drone instrument), a singing voice signal, and a conga rim-hit signal (drum family). Furthermore the algorithm has been applied to some other plucked string signals such as a bass and a pizzicato cello signal.

3.6 Results and Discussion

This section presents the simulation results of the proposed algorithm. The results of the main guitar test signal are presented first. The results using the Haar wavelet, D4-wavelet and the Morlet wavelet are also described. Next the results of

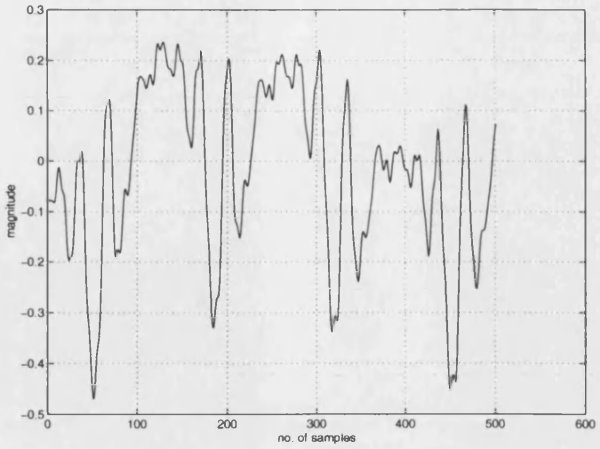

Figure 3.11: A segment of the guitar signal

the other musical signals are presented. Finally a comparative discussion between the proposed algorithm and the autocorrelation function is presented.

3.6.1 The Analysis of the Guitar Signal

Generally in the analysis of plucked string signals, the results show that it is sufficient to estimate the pitch period from the steadily decaying part of the signal several hundred milliseconds after the attack [55]. This is due to the fact that the pitch period of a plucked-string signal decreases as the signal attenuates. In the analysis of the E2-tone signal, shown in figure (3.2), the first 10000 samples have been discarded which amounts to 227 ms. A segment of the guitar signal of 500 samples is shown in figure (3.11). For the quadratic spline wavelet the estimated pitch is 330.3293 Hz. The results are shown in figure (3.12).

The performance of the other wavelets is found to be the same for this signal. Furthermore the smoothness of the minimum phase wavelet is investigated where a D20-wavelet [8] is used in the analysis. The results show that the smoothness of the analysis wavelet is not a crucial parameter.

The E2-tone is the main analysis test signal. However, the proposed algorithm

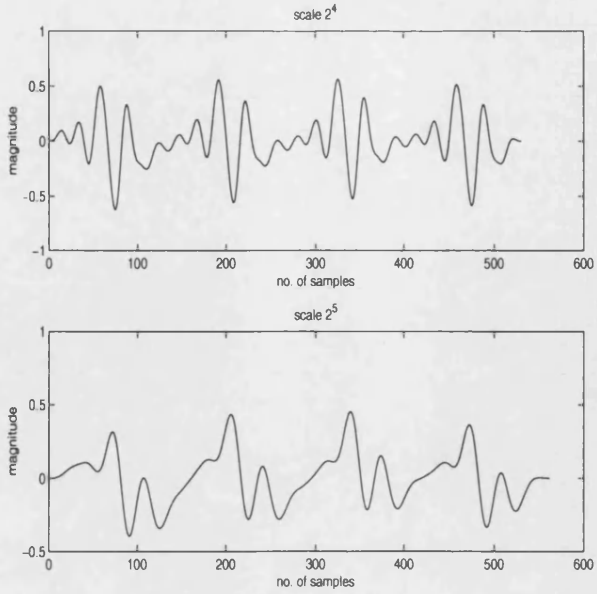

Figure 3.12: Scales 2^4 and 2^5 of the guitar signal using the quadratic spline

Tone	True Pitch	Estimated pitch
E3	661.8095	662.0249
G1	196.2381	197.6210

Table 3.3: Estimated pitch period of various guitar signals

has been applied to a wide range of guitar signals. Figures (3.13) and (3.14) show the results of two other guitar signals. The estimated pitch periods are shown in table (3.3).

The performance of the algorithm has been also tested for a synthetic guitar signal with two pitch periods of 100Hz and 200Hz. The guitar signal and the results are shown in figure (3.15). The estimated pitch period using all test signals is 99.4222 Hz, i.e. the difference between the two pitch periods. This can be justified as the guitar signal having a missing fundamental of 100 Hz with 200 Hz and 300Hz as the second and the third partials. Nevertheless this case is open for further investigation.

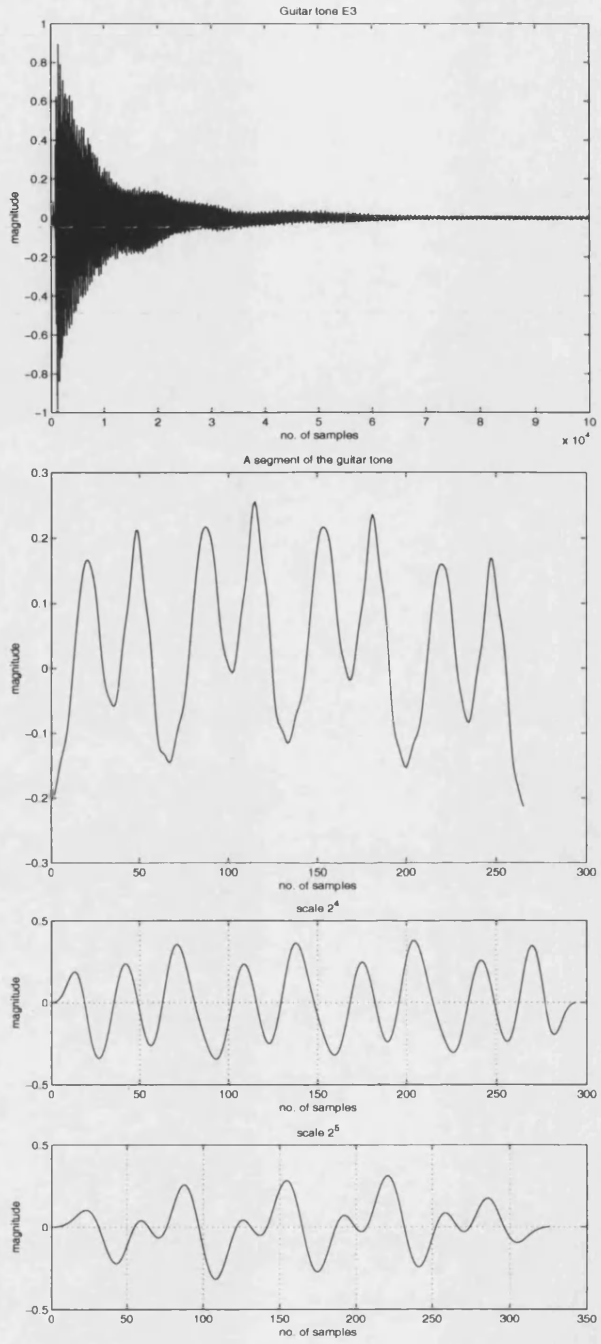

Figure 3.13: Results of the guitar signal E3

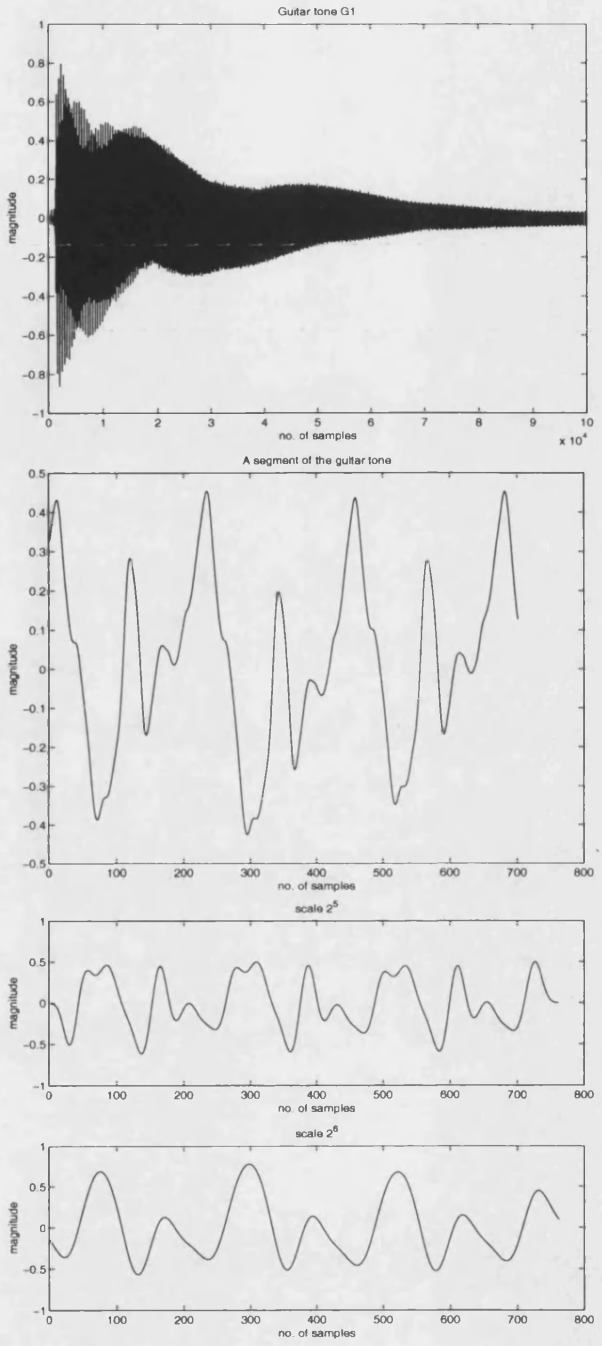

Figure 3.14: Results of the guitar signal G1

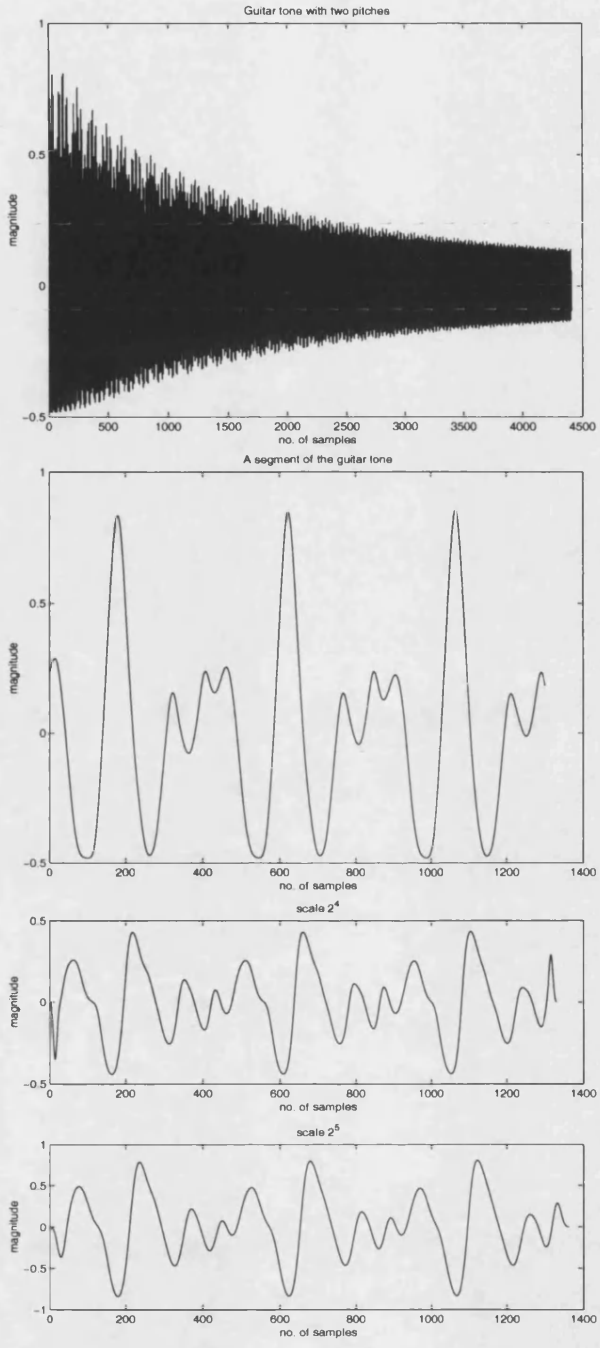

Figure 3.15: Results of a guitar signal with two pitch periods

3.6.2 The Analysis of other Musical Signals

The true pitch period of a D#saxaphone signal is 155.5144Hz. The estimated period is 154.7368Hz with relative error 0.005 where the relative error (r.e.) is defined as

$$\text{r.e.} = \text{abs(true pitch - estimated pitch)/true pitch}$$

For all used wavelets the estimated pitch is the same.

The tanpura signal is a very harmonically rich signal. Our results show that the proposed algorithm has the ability to detect not only the fundamental frequency but also the frequency with the most energy present in the signal. In the case of the autocorrelation function a longer frame size has to be used in order to detect the fundamental of the tanpura signal not the strongest harmonic [10]. The estimated pitch period of this signal is 157.5Hz.

In the analysis of a male singing voice, the estimated pitch is 110.8040Hz. The autocorrelation function gives slightly different values depending on the analysis window as well as the frame size. Moreover in the analysis of a conga-rim signal, the algorithm classified this signal as an unpitched one since it failed to find local maxima that satisfy the previous criteria. The results of all test signals are shown in [10] and in appendix (A) and further explained in [46].

3.6.3 Discussion

The computational complexity of the proposed algorithm is O(NJ), for an input signal of length N evaluated across J scale. The constant depends on the number of nonzero coefficients present in the filters h and g. The algorithm is faster than the autocorrelation function since the length of the analysis wavelet is less than M_0.

Unlike the autocorrelation function the proposed algorithm takes into account the non-stationary nature of musical signals. Besides the window function is not a crucial parameter since the quadratic spline wavelet can be used for all kinds

of musical signals. Furthermore the frame size is not a crucial parameter since different frame sizes have been successfully used in the analysis. It is worth mentioning here that all the signals used are not professionally recorded and are subjected to one sort of noise or another. This implies that the proposed algorithm is robust to noise. It is also accurate if we use the relative error criteria as our measure of accuracy.

3.7 Summary

In this chapter we have presented a new algorithm for estimating the pitch periods of musical signals. The proposed algorithm is based on the *dyadic wavelet transform* and a quadratic spline wavelet with certain criteria has been constructed for this purpose. This chapter has also presented the main advantages of the proposed algorithm over the well-known autocorrelation function. Further the effect of the analysis wavelet has been tested by using different wavelets and the quadratic spline wavelet has proven to be suitable for all test signals.

Chapter 4

Kalman Filtering Technique for Partials Tracking

4.1 Introduction

As described in the previous chapter the objective of the analysis process is to provide the data required for calibrating the guitar model under consideration. The first step of the analysis process is estimating the pitch period of a recorded real guitar tone, which was presented in the previous chapter.

In this chapter the analysis process continues by studying the time-varying spectrum of the guitar tone. There are different tools for the analysis of this spectrum. The "heterodyne filter" was developed by Moorer in 1973 [42]. This filter implements a single frequency bin of the Discrete Fourier Transform (DFT) using the rectangular window. The instantaneous amplitude and frequency are estimated from the magnitude and phase derivative of the complex numbers produced by the sliding DFT. This technique was developed further by using the digital "phase vocoder" where the Fast Fourier Transform (FFT) is used to implement the DFT along with the use of a non-rectangular window [42]. The phase vocoder is useful for the analysis of harmonic sounds whereas sounds with time-varying characteristics were difficult to analyse using this technique. In 1987, Smith developed another peak-tracking spectral analyzer, called PARSHL [52]. This technique is based on the Short-Time Fourier Transform (STFT) [2] [1] and can be applied to in-harmonic sounds such as the piano. However, PARSHL has difficulty representing noise-like signals such as the attack part of many musical signals. In 1986, McAulay and Quatieri developed a similar technique for the analysis of speech signals [31]. In musical applications, this technique was developed further by Serra and Smith [45]. In 1990, they proposed the Spectral Modelling Synthesis (SMS) where musical signals can be represented as the sum of sinusoids and noise. This scheme is also built on top of the STFT. The basic idea adopted is that the guitar signal can be represented as a set of quasi-periodic stable partials (sinusoidal model) that correspond to the main modes of vibration of the string, and some added noise (the residual model) [55]. Each partial

feature such as amplitude, frequency and phase values, has to be extracted. The decay rates of such partials determine the magnitude responses of the loop filter, as will be further discussed in the next chapter. Once the trajectories of all distinctive partials have been computed, the sinusoidal model is then synthesised using additive synthesis whereas the residual model is obtained by subtracting the sinusoidal model from the original signal. Thereafter the residual model will be post-processed to extract the excitation signals for the guitar model.

Our goal in this chapter is to present a new technique for partial tracking of musical signals. The proposed technique is based on the theory of the Extended Kalman filter [5]. The main principle adopted here is that the guitar signal can be represented as the sum of a deterministic model (sinusoidal model) and a stochastic model (residual model). Each partial of the sinusoidal model is modelled using three different state variables. Since the Kalman filter is a model-based technique, several models have been tested for our new technique. Different aspects of the implementation of the proposed technique are illustrated. Furthermore the advantages and disadvantages of the proposed technique are presented.

This chapter is organised as follows: section 2 is devoted to the Short-Time Fourier Transform (STFT) as an analysis tool for non-stationary signals. The analysis of the real guitar tone is carried out using this technique to extract the features of each partial. Some implementation aspects are also briefly described. In section 3, we introduce the mathematical definitions and notations which are needed for understanding the concept of the Kalman filter. The discrete-time Kalman filter is then described. The non-linear case is subsequently addressed with emphasis on the Extended Kalman filter technique. Different models that are used in the analysis are presented in section 4. Advantages and critiques of each model are also addressed. Section 5 presents the implementation aspects of the proposed algorithm, the results of the new algorithm being presented in section 6. Comparisons are also made with those derived from the Short-Time

Fourier analysis. Finally the chapter summary is given in section 7.

4.2 Short-Time Fourier Transform-based Analysis

The purpose of this section is to estimate the time-varying spectrum of the recorded real guitar tone using the Short-Time Fourier Transform (STFT). We start by describing the principle of the STFT and its implementation and also consider the control parameters for the computation of the STFT. We then discuss the use of this transform for estimating the partial trajectories of the guitar signal.

4.2.1 The Short-Time Fourier Transform

Musical signals are non-stationary signals, i.e they evolve with time. In order to study the spectrum characteristics of such signals, the usual Discrete Fourier Transform (DFT) is not appropriate due to this non-stationary nature. Instead a time-dependent version of the DFT, known as the Short-Time Fourier transform (STFT), is utilised. It is based on the idea that a time-domain window, usually zero outside a certain interval, is moving or 'hopping' along the signal to be analysed. For each windowed segment, one computes the DFT which represents the spectrum of this segment. In practice the DFT can be implemented efficiently using the Fast Fourier Transform (FFT) when the size of the signal is restricted to a power of two [36], [38], and [40].

For a discrete-time signal $x(n)$, the STFT is defined [24] as

$$X_l(k) = \sum_{n=0}^{M-1} w(n)x(n + lH)e^{-i\omega_k n},\ for\ l = 0, 1, 2 \ldots \tag{4.1}$$

where $w(n)$ is an appropriate window function, M is the FFT size, $k = 0, 1, \ldots, M-1$ is the bin index, $\omega_k = 2\pi k/M$ is the radian frequency, n is the time index param-

eter, and H is the hop size parameter, defined as the time advance (in samples) between the consecutive frames.

The result of STFT is a set of frames separated in time by the hop size. Each frame is represented by a complex-valued function called the spectrum which represents the amplitude and phase spectrum when transformed to polar coordinates. The spectral partials are represented in each frame as peaks in the spectrum. These peaks can be detected and their frequency, amplitude and phase evaluated, as will be described in the rest of this section. It is clear that the control parameters for STFT are the window function and its size, the hop size, and the FFT size. These parameters must be determined for each signal to be analysed.

4.2.2 Control Parameters for the STFT

In this section we discuss the control parameters for the STFT in more detail.

The choice of an appropriate *window function* is a compromise between the frequency resolution and the time resolution [14], [45]. A window function with a narrow main-lobe (i.e. better frequency resolution) and a very low side-lobe would be an ideal choice for many applications. A suitable window for the purpose of this analysis is the Hamming window, where the highest side-lobe is 43 dB below the main-lobe [35]. The Hamming window is defined as:

$$w(n) = \begin{cases} 0.54 - 0.46 * \cos\left(\frac{2\pi n}{N-1}\right) & \text{for} \quad n = 0, 1, 2, \ldots, N-1 \\ 0 & \text{elsewhere} \end{cases}$$

where N is the window size. A graph of the Hamming window is shown in figure (4.1)

The *window size*, N, is another crucial parameter in the analysis. The window size determines the time resolution of the analysis. A smaller window results in a better time resolution (but worse frequency resolution) and vice versa. Since the purpose of the analysis is to track closely-related partials, frequency resolution is

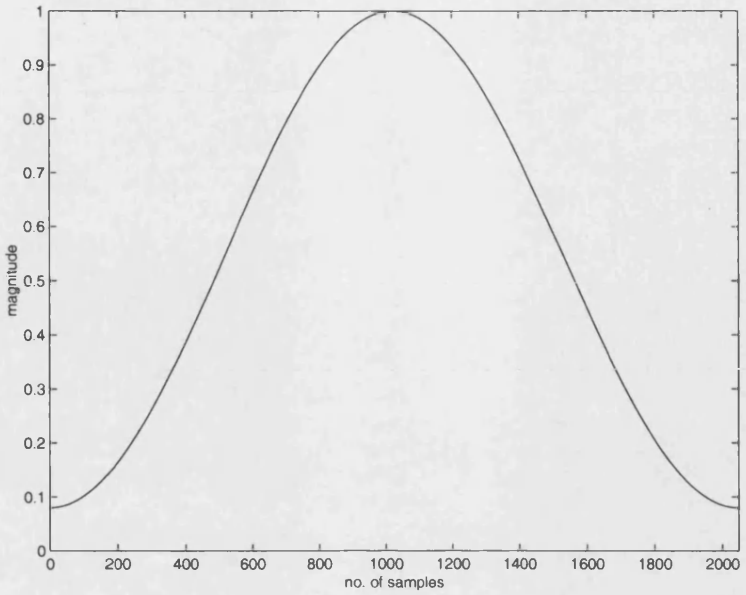

Figure 4.1: The Hamming window with N=2048

favoured over time resolution. For pitched sounds, McAulay and Quatieri have reported that the window size should be at least two and a half times the pitch period [31]. In our implementation, the window size has been set to four times the pitch period.

Since the window size could be any integer (not necessarily a power of two), depending on the estimated pitch period, and also in order to enable the use of the FFT in the analysis, *zero-padding* is used [45]. Zero-padding is achieved by inserting zeros in the windowed segment buffer until the FFT size, M, is the nearest power of two which is at least twice the window size N. The importance of zero-padding lies not only in the ability to use FFT, but also in increasing the frequency resolution of the analysis. In other words, zero-padding in the time domain is equivalent to frequency interpolation in the frequency domain [45].

Another crucial parameter in the implementation of STFT is the *hop size* parameter H. The choice of the hop size parameter is very much application-dependent. A smaller hop size implies a smoother spectrum at the expense of

higher computational cost. For a Hamming window, a good choice for H is one fourth of the window size [45], [1]. This means that there is 75% overlap between consecutive frames.

A further technique which may be used to improve the accuracy of the estimated phase values is *zero-phase windowing* [45], [55], [29]. The importance of the accuracy of the estimated phase values stems from the fact that the residual signal is obtained from the original signal by subtracting the sinusoidal model in the time domain.

Zero-phase windowing can be implemented in practice by using the circular properties of the DFT, i.e. the windowed signal is considered to be periodic with period N (which is the window size). So instead of applying the DFT to the windowed segment from $[0, N-1]$, we use the segment from $[(N+1)/2, (3N-1)/2]$. In such a case, zero-padding is implemented by inserting zeros in the middle of the zero-phase windowed segment [55]. By using zero-phase windowing, a linear trend in the phase values, due to the windowing process, is avoided. It is worth mentioning here that the window size must be an odd number for zero-phase windowing to be applicable. In figure (4.2) a portion of the test signal is windowed using a Hamming window of length $N = 537$. Zero-phase windowing is then utilised (as well as zero-padding) to a length of $M = 4096$ to increase the frequency resolution.

The magnitude and phase spectrum of this zero-phase, zero-padded windowed segment are shown in figure (4.3). This is a typical frame in the STFT analysis. It is clear that the phase spectrum is nearly a constant value in the vicinity of a harmonic due to zero-phase windowing. This results in reducing the error of the estimated phase values.

Figure (4.4) shows a 3D-plot of the STFT of the E2-guitar test signal, shown in figure (3.2), showing the time-evolution of the signal partials.

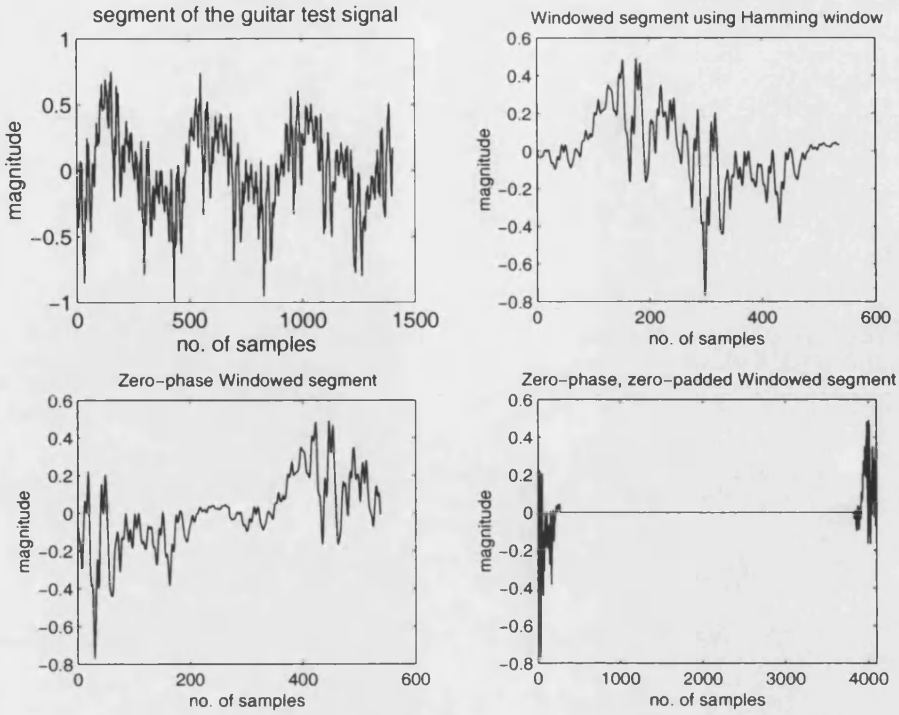

Figure 4.2: Zero-phase, zero-padded windowed guitar segment

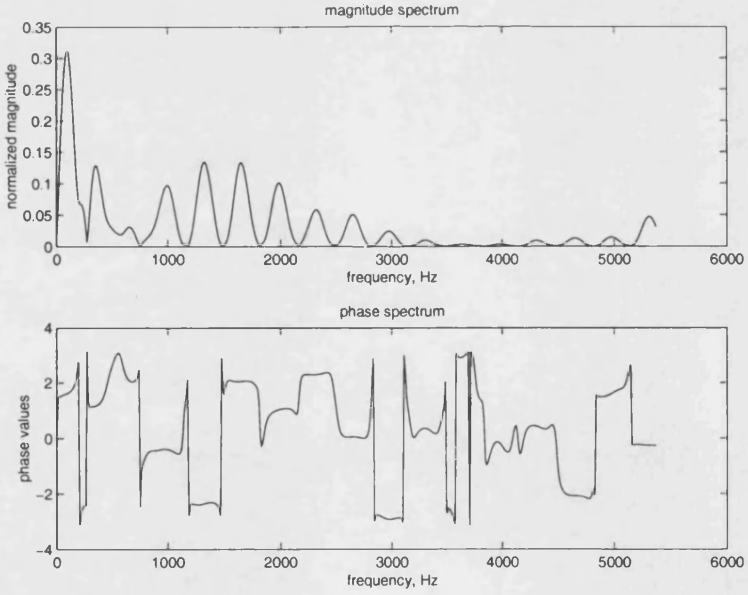

Figure 4.3: Magnitude and phase spectrum of a typical frame in the STFT

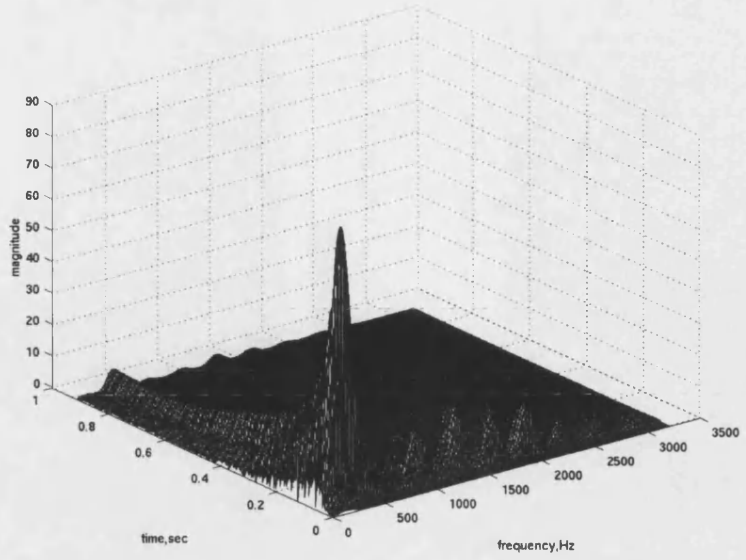

Figure 4.4: 3D-plot of the STFT of the guitar signal

4.2.3 Peak Detection and Continuation

The next step in the analysis process is to detect the peaks present in the spectrum. These peaks amount to the prominent spectral components of the guitar signal. This step is accomplished by using a peak detection algorithm, which is performed on a frame-by-frame basis by exploiting the nearly-harmonic nature of the guitar tone.

Performed on a dB-scale, this algorithm can be summarised as follows:

- For each frame l and every partial r, search for the maximum value in the magnitude spectrum within a certain frequency deviation f_D, i.e. in the interval $[rf_0 - f_D, rf_0 + f_D]$, where f_0 is the estimated pitch period for the guitar signal. In the current implementation, a value of $f_D = f_0/4$ has been found to be appropriate.

- If this maximum value corresponds to a local maximum, i.e. if

$$|X_l(k)| > |X_l(k-1)|,$$
$$|X_l(k)| \geq |X_l(k+1)|$$

where $|X_l(k)|$ is the magnitude spectrum of frame l at bin index k, then this maximum value is marked as a peak.

- The index, amplitude, frequency and phase values of such a peak are computed.

- If the amplitude value of such a peak is less than a certain magnitude threshold, this peak is ignored, i.e. a zero value is assigned to its magnitude. This step checks the validity of the detected peaks so that irrelevant peaks can be rejected. In our experiment, results show that a magnitude threshold of -70 dB has been found to be appropriate in our implementation (before the signal reaches the noise floor).

- The detected magnitude values must be normalised in order to compensate for the windowing effect. Hence the magnitude spectrum of each frame is multiplied by a normalisation factor c_N given by:

$$c_N = \sum_{n=0}^{N-1} \frac{2}{w(n)}$$

- The frequencies and normalised magnitudes can be further improved by using parabolic interpolation [55], as previously described in chapter (3).

After scanning all frames for valid peaks, a peak continuation algorithm is utilised in order to provide for each partial's trajectory for the sinusoidal part of the guitar signal. The basic idea of this algorithm is to link the valid peaks between frames, assuming that the partials are fairly stationary between frames. If we assume that we are currently at frame l, the continuation algorithm can be simplified as follows:

Parameter	Value (samples)
window size	537
FFT size	4096
hop size	134

Table 4.1: Values of the STFT control parameters

- If a peak exists in frame $l+1$ that corresponds to an existing trajectory within a certain frequency deviation f_D, then the peak is said to be *advanced* in time. The detected frequency, amplitude and phase values of this peak are assigned to the existing trajectory.

- If there is no peak that corresponds to the existing trajectory within the assumed frequency deviation, then the peak is said to be *killed* entering frame $l+1$. In such case, a triplet consists of zero amplitude, the same frequency as in frame l and the same phase value, plus a phase shift equal to $d\phi = \omega_r H$.

- If there is a valid peak in the current frame that does not correspond to an existing trajectory, then a trajectory is *born*. A triplet is created in the previous frame with zero amplitude, the same frequency (of the peak) and the same phase of the peak minus a phase shift equal to $d\phi = \omega_r H$.

Table (4.1) gives the values of the STFT control parameters used for the analysis of the guitar test signal. The analysis window is the Hamming window. Figure (4.7) shows the amplitude trajectories $E_r(l)$ of different partials of the guitar signal. It is clear that some of them are decaying exponentially. It is also clear from the figure that the second harmonic is very small compared to the other partials.

For more detailed discussion about the peak detection and the peak continuation algorithm, we refer the reader to [31], [45] and [55].

4.2.4 Problems with the STFT

The STFT is a powerful tool for the analysis-synthesis of musical signals. However there are some limitations and disadvantages to this algorithm. The main problem associated with the STFT is the resolution of the analysis, which is directly related to the window size. According to the Heisenberg uncertainty principle, one can not know exactly what frequencies are present in a signal at a given instant of time. What one gets is a band of frequencies present in an interval of time. There is thus always a trade-off between time and frequency resolution.

In addition to the resolution problem there is also the assumption that the signal is stationary within each windowed segment. The computational complexity is another problem associated with the STFT as will be further explained in the following chapters.

4.3 The Discrete Kalman Filter

This section serves as a basic introduction to the Kalman filter. It starts by describing the mathematical notation that is necessary to understand the principles of this filter. The discrete linear Kalman filter is then briefly presented. The section continues by describing the nonlinear model, in particular the extended Kalman filter model.

4.3.1 Mathematical Notation

The purpose of this section is to introduce some basic mathematical definitions that are often used in the terminology of the Kalman filter [5].

Given a random variable x, the *variance* of x is defined as

$$var(x) = \sigma^2 = E[(x - E(x))^2] \tag{4.2}$$

where $E(x)$ is the expectation of x, i.e. the mean or average value of x. In a qualitative sense, the variance of x is a measure of the dispersion of x about the mean value $E(x)$. Whenever the mean value is zero, the variance is just the second moment.

Equation(4.2) can be rewritten as

$$var(x) = \sigma^2 = E(x^2) - (E(x))^2 \tag{4.3}$$

The positive square root of the variance is called the *standard deviation.*

A random variable x is said to be *normal* or *Gaussian* if its probability density function is given by

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-m_x)^2} \tag{4.4}$$

where m_x and σ^2 are the mean and variance of the random variable x respectively. In what follows, the notation

$$x \sim N(m_x, \sigma^2)$$

is used as a shorthand notation for a normal random variable.

The *covariance* of two random variables x and y is defined as

$$cov(x, y) = E[(x - m_x)(y - m_y)] \tag{4.5}$$

It is a measure of how x and y co-vary.

Consider a set of n random variables $x_1, x_2, \ldots, x_n$ (also called variates). A *vector random variable* $\mathbf{x}$ is defined as

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

In general, the components of $\mathbf{x}$ may be correlated and have nonzero means. The *covariance matrix* for $\mathbf{x}$ is defined as

$$C = \begin{bmatrix} var(x_1) & cov(x_1, x_2) & \cdots & cov(x_1, x_n) \\ cov(x_2, x_1) & var(x_2) & \cdots & cov(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(x_n, x_1) & \cdots & \cdots & var(x_n) \end{bmatrix}$$

The terms along the major diagonal of the covariance matrix are the variances of the variates whereas the off-diagonal terms are their covariances. The covariance matrix C of a vector random variable $\mathbf{x}$ describes the variances and correlation structure of the n-variate.

A *white noise* is defined to be a stationary random process having a constant spectral density function.

4.3.2 The Discrete-Time Kalman Filter

The Kalman filter is a computational algorithm that provides an efficient solution of the least-squares method. Its purpose is to estimate the state of a process from noisy measurements. The Kalman filter has a wide range of application areas. It has been proven to be useful in navigational and guidance systems, radar tracking, sonar ranging and satellite orbit determination. In music applications, the Kalman filter has been used in audio restoration of electro-acoustic music [37], polyphonic-sources separation [63], and in the identification characteristics of woodwind instruments [6]. In this section, we briefly introduce the discrete, linear Kalman filter [5], [4], and [30].

Consider a discrete-time random process $\mathbf{x}$, which is to be estimated, governed by the linear equation:

$$\mathbf{x}_{k+1} = \phi_k \mathbf{x}_k + \mathbf{w}_k \tag{4.6}$$

where

$\mathbf{x}_k$ – is an $n \times 1$ process state vector at time t_k,

ϕ_k – is the $n \times n$ state transition matrix, and

$\mathbf{w}_k \sim N(0, Q_k)$ – is the process noise.

The state transition matrix ϕ_k relates the state of the process at two time instants t_k and t_{k+1} in the absence of any noise. The measurements (or observations) of the process are assumed to occur at discrete instants of time $t_1, t_2, \ldots$, according to the linear equation

$$\mathbf{y}_k = H_k \mathbf{x}_k + \mathbf{v}_k \tag{4.7}$$

where

$\mathbf{y}_k$ is an $m \times 1$ measurement vector at time t_k,

H_k is an $m \times n$ measurement matrix, and

$\mathbf{v}_k \sim N(0, R_k)$ is the measurement noise.

Both $\mathbf{v}_k$ and $\mathbf{w}_k$ are assumed to be white noises of zero mean and normal distribution. Furthermore, we assume that their covariance structures are known, i.e

$$E(w_k\, w_i^T) = \begin{cases} Q_k, & \text{if } i = k \\ 0, & \text{otherwise} \end{cases}$$

$$E(v_k\, v_i^T) = \begin{cases} R_k, & \text{if } i = k \\ 0, & \text{otherwise} \end{cases}$$

and

$$E(v_k\, w_i^T) = 0, \quad \text{for all } i \text{ and } k$$

where T stands for the transpose of the vector. Now, assume that we have an initial estimate $\hat{\mathbf{x}}_k$ of the state $\mathbf{x}_k$ at this point in time t_k. This estimate is based

on our knowledge about the process prior to t_k. Further, assume that the error covariance matrix $\hat{P}_k$ associated with this estimate is known. Such a matrix is given by

$$\begin{aligned} \hat{P}_k &= E[e_k e_k^T] \\ &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \end{aligned}$$

where e_k is the estimation error. Using the measurement $\mathbf{y}_k$, we seek to update our initial guess according to the equation

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + K_k[\mathbf{y}_k - H_k\hat{\mathbf{x}}_k] \tag{4.8}$$

where K is an $n \times m$ matrix called the *Kalman gain* or the *Kalman factor* and is chosen in order to minimise the error variance. The Kalman gain is given by

$$K_k = \hat{P}_k H_k^T (H_k \hat{P}_k H_k^T + R_k)^{-1} \tag{4.9}$$

The updated error covariance matrix associated with equation (4.8) is

$$P_k = (I - K_k H_k)\hat{P}_k \tag{4.10}$$

where I is the identity matrix. The algorithm is projected ahead via the following two equations

$$\hat{\mathbf{x}}_{k+1} = \phi_k \mathbf{x}_k \tag{4.11}$$

$$\hat{P}_{k+1} = \phi_k P_k \phi_k^T + Q_k \tag{4.12}$$

Now the required quantities for the next time step t_{k+1} are known as well as the measurement $\mathbf{y}_{k+1}$. The last five equations comprise the Kalman filter algorithm and are summarised in figure (4.5). After a while the algorithm starts to depend more on the measurements and less on the initial values.

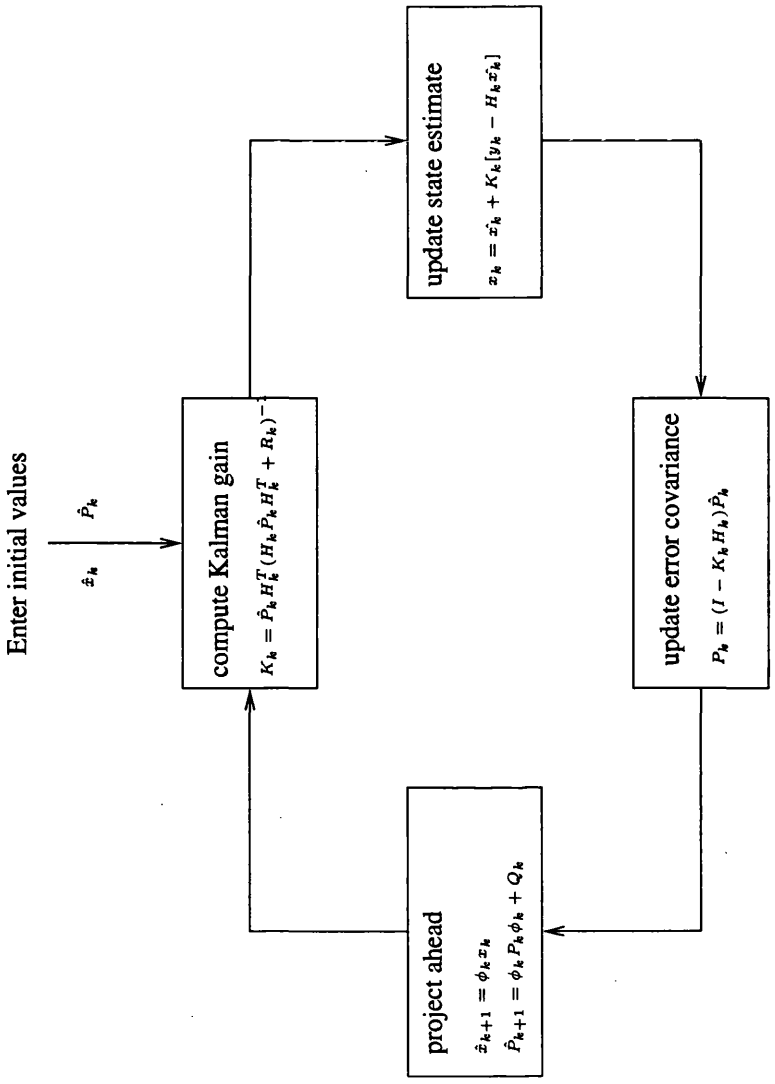

Figure 4.5: The Kalman filter loop

4.3.3 The Extended Kalman Filter

In the previous section we presented the principles of the linear Kalman filter case. In this case the Kalman filter is considered to be the optimal linear estimator. In real life, however, there are many situations where the state process matrix and/or the measurement matrix are nonlinear. The technique employed in these situations is to linearise the model in order to fit into the linear case. There are two ways of linearisation. One is to linearise about some nominal trajectory in the state space that does not depend on the measurements. Such a technique is known as the *linearised Kalman filter.* The second method is to linearise about a trajectory that is continuously updated with the state estimates resulting from the measurements. In such a case the filter is known as the *Extended Kalman filter*(EKF). In what follows the equations of the EKF are briefly described.

Consider the case when both the state process matrix and the measurement matrix are both non-linear functions in accordance with the equations

$$\mathbf{x} = \mathbf{f}(\mathbf{x}, t) + \mathbf{w}(t) \tag{4.13}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, t) + \mathbf{v}(t) \tag{4.14}$$

where $\mathbf{f}$ is a vector function of size $n \times 1$. Each element $\mathbf{f}_i$ is a function of the state variable vector $\mathbf{x}$ of n-variates. Also $\mathbf{h}$ is a vector function of size $m \times 1$, where $\mathbf{h}_i$ is again a function of the vector state variable $\mathbf{x}$. Both $\mathbf{w}$ and $\mathbf{v}$ are as defined previously, and $\mathbf{y}$ is the measurement matrix of size $m \times 1$.

If we have an initial estimate $\hat{\mathbf{x}}_k$ for the state process vector $\mathbf{x}$ at a time instant t_k, then both $\mathbf{f}$ and $\mathbf{h}$ can be expanded about this estimate using Taylor series. For small values, the second and higher orders of the series are negligible.

In the non-linear case, the state process matrix ϕ is defined as

$$\phi = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \tag{4.15}$$

$$= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{4.16}$$

and the measurement matrix H is given by

$$H = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \tag{4.17}$$

$$= \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial x_1} & \frac{\partial h_m}{\partial x_2} & \cdots & \frac{\partial h_m}{\partial x_n} \end{bmatrix} \tag{4.18}$$

Hence the EKF recursive equations can be implemented as follows [12]

- Compute the Kalman filter gain

$$K_k = \hat{P}_k H_k^T (H_k \hat{P}_k H_k^T + R_k)^{-1} \tag{4.19}$$

  where $\hat{P}_k$ is the initial estimate of the error covariance matrix associated with the initial estimate $\hat{\mathbf{x}}_k$ (as previously described).

- Update both the estimated state process and the error covariance matrix according to the following relations

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + K_k[\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k)] \tag{4.20}$$

$$P_k = (I - K_k H_k)\hat{P}_k \tag{4.21}$$

- Project ahead to the next step via the equations

$$\hat{\mathbf{x}}_{k+1} = \phi_k \mathbf{x}_k \tag{4.22}$$

$$\hat{P}_{k+1} = \phi_k P_k \phi_k^T + Q_k \tag{4.23}$$

Note that both ϕ and H are obtained by evaluating the first partial derivative of the vector functions $\mathbf{f}$ and $\mathbf{h}$ along their respective trajectories.

4.4 Modelling

In this section we present the underlying model for parameter estimation of the guitar signal. The Kalman filter depends mainly on the choice of an appropriate model for the estimation process. This choice must be a compromise between simplicity and accuracy in describing the problem in hand. Therefore, different models have been tested for the task of parameter estimation of the guitar signal. We start this section by presenting the initial model that has been suggested. Critiques of this model are then discussed. A modified and final model is presented along with its main advantages over the initial one.

4.4.1 The Initial Model

Again we follow the same assumption that has been used before in the STFT: that the guitar signal can be represented as a sum of a stochastic model (sinusoidal) and a noise model (residual). In the ideal case, each partial should decay exponentially. Hence each partial of the guitar signal can be expressed as

$$y_k(t) = A_k e^{-B_k t} \cos(\omega_k t + \theta_k) \tag{4.24}$$

where

k is the partial number,

θ_k is the phase offset,

A_k is the partial amplitude at zero time,

B_k is the decay rate of the partial.

Expanding the cosine function and substituting for the appropriate state variables, we have

$$y_k(t) = x_{1k}x_{3k}\cos(\omega_k t) - x_{2k}x_{3k}\sin(\omega_k t) \tag{4.25}$$

where the state variables of this partial are given by

$$\begin{bmatrix} x_{1k} \\ x_{2k} \\ x_{3k} \end{bmatrix} = \begin{bmatrix} A_k\cos\theta_k \\ A_k\sin\theta_k \\ e^{-B_k t} \end{bmatrix}$$

Thus each partial is represented by three state variables. These variables determine its amplitude and its phase offset, as well as its rate of decay.

The guitar signal can therefore be expressed as

$$y(t) = \sum_{k=1}^{k=K} y_k(t) + \mathbf{v}(t) \tag{4.26}$$

where K is the maximum number of partials present in the guitar signal and $\mathbf{v}(t)$ is a Gaussian white noise that represents the residual model.

This implies that the measurement is related to the state by the non-linear function $\mathbf{h}$ given by

$$\mathbf{h} = \sum_{k=1}^{K} x_{1k}x_{3k}\cos(\omega_k t) - x_{2k}x_{3k}\sin(\omega_k t) \tag{4.27}$$

Hence the measurement transition matrix H, which is a $1\times 3K$ vector, is evaluated as

$$H = \begin{bmatrix} H_1 & H_2 & \cdots & H_K \end{bmatrix} \tag{4.28}$$

where

$$H_i = \begin{bmatrix} x_{3i}\cos(\omega_i t) & -x_{3i}\sin(\omega_i t) & \Big(x_{1i}\cos(\omega_i t) - x_{2i}\sin(\omega_i t)\Big) \end{bmatrix} \tag{4.29}$$

for $i = 1, 2, \ldots, K$.

In the discrete domain, the state transition matrix in this case evaluates to

$$\phi = \begin{bmatrix} \phi_1 & Z & \cdots & Z \\ Z & \phi_2 & \cdots & Z \\ \vdots & \vdots & \ddots & \vdots \\ Z & Z & \cdots & \phi_K \end{bmatrix} \tag{4.30}$$

where Z is a zero matrix of order 3 and

$$\phi_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-B_i \Delta t} \end{bmatrix} \tag{4.31}$$

for $i = 1, \ldots, K$, and Δt is the sampling interval.

Figure (4.6) shows some estimated state variables of various partials using the initial model.

4.4.2 Initial Model Critiques

Two major problems with this model can be noted. Firstly, the assumption that the partials are decaying exponentially does not hold in practice. This is due to non-linearities present in the true signal such as beats and two-stage decay. The initial model does not take this fact into account, which results in an incorrect estimation of some partials.

Secondly, the initial model also assumes the stationarity of the frequency of each partial (since frequency is not one of the state variables of each partial). This assumption again violates the nature of a musical signal where there is some sort of frequency modulation that needs to be taken into account.

4.4.3 The Final Model

Our objective is to build a model that takes into account the two-stage decay as well as the frequency and amplitude modulation of the guitar signal. A more

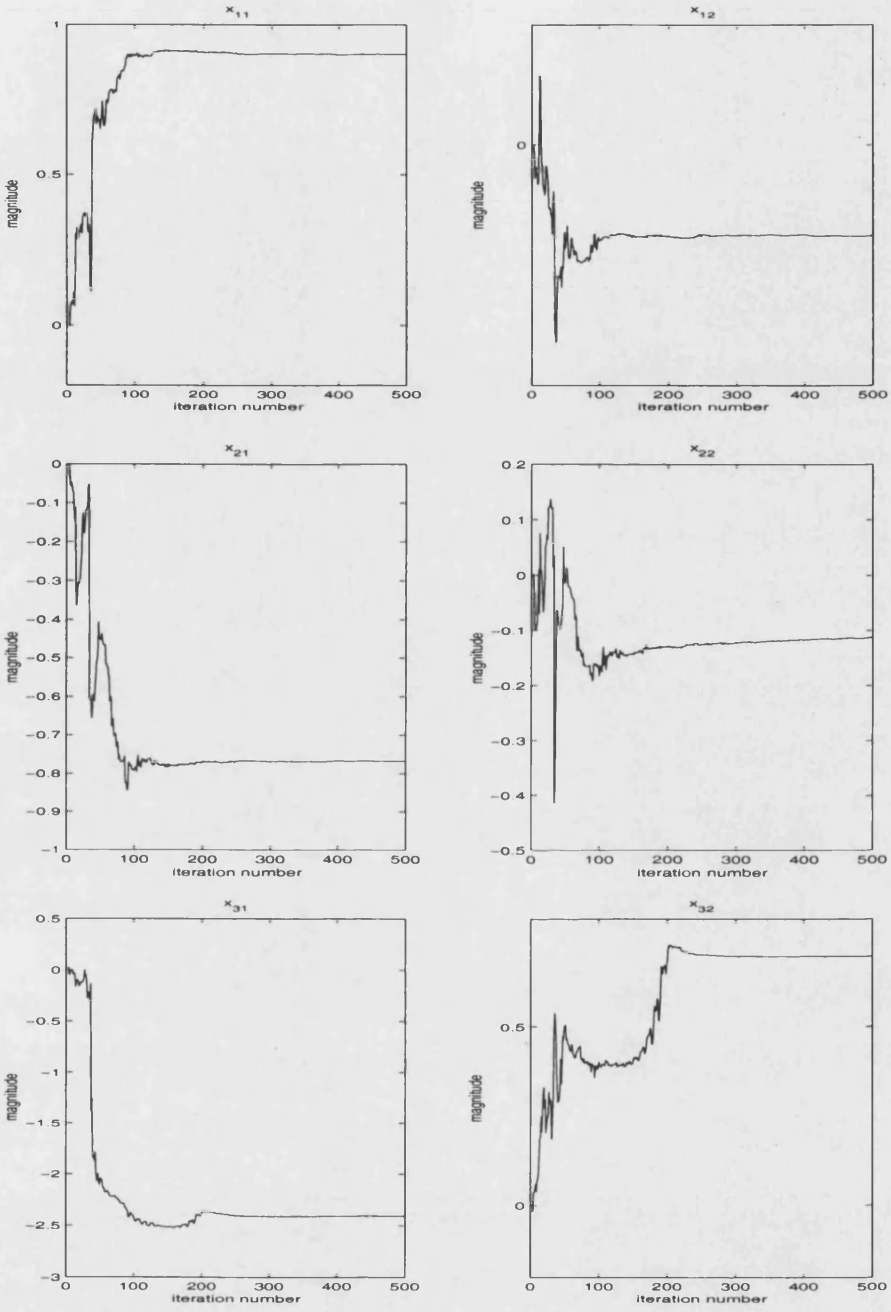

Figure 4.6: Various state variables using the initial model

realistic model represents each partial as

$$y_k(m) = A_k(m)\cos\left(\theta_k(m)\right) \tag{4.32}$$

where $A_k(m)$ is the time-varying amplitude of the k^{th} partial and $\theta_k(m)$ is its instantaneous phase. The instantaneous phase is related to the instantaneous frequency $f_k(t)$ through the following relation

$$\theta_k(m+1) = 2\pi f_k(m)\Delta t + \theta_k(m) \tag{4.33}$$

Hence the guitar signal can be expressed as

$$y(m) = \sum_{k=1}^{k=K} A_k(m)\cos\left(\theta_k(m)\right) + \mathbf{v}(m) \tag{4.34}$$

In this case we define three state variables for each partial, namely the amplitude, the phase and the frequency. That is

$$\begin{bmatrix} x_{1i} \\ x_{2i} \\ x_{3i} \end{bmatrix} = \begin{bmatrix} A_i(m) \\ \theta_i(m) \\ f_i(m) \end{bmatrix}$$

for $i = 1, 2, \ldots, K$.

In order to take into account the time-varying nature of the frequency and amplitude of each partial, both x_{1k} and x_{3k} are modelled as random walks. Hence the state transition matrix ϕ for this model is defined as in equation (4.30), with ϕ_i defined as

$$\phi_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2\pi\Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{4.35}$$

for $i = 1, \ldots, K$. The measurement matrix H is defined as in equation (4.28), with H_i defined as

$$H_i = \begin{bmatrix} \cos\left(x_{2i}\right) & -x_{1i}\sin\left(x_{2i}\right) & -x_{1i}\sin\left(x_{2i}\right)2\pi\Delta t \end{bmatrix} \tag{4.36}$$

It is clear that the measurement matrix H is not a time-invariant matrix. Therefore it has to be computed for each iteration step in the EKF loop.

4.5 Simulation

The objective of the simulation is to provide the best estimate of the vector state variable $\mathbf{x}_k$. This vector describes a certain dynamic process at a time instance t_k given a set of noisy measurements $y_1, y_2, \ldots, y_k$. In this section we present the different implementation aspects of the proposed algorithm. It also describes the various techniques that are utilised to optimize the performance of the proposed algorithm.

4.5.1 Initialisation

The control parameters of the proposed algorithm are: the initial estimate of the state vector, the associated error covariance matrix, the total number of partials, the state covariance matrix, and the sampling interval. In order to start the recursive equations of the EKF algorithm, an initial estimate of the vector state variable $\hat{\mathbf{x}}_k$ at time instance t_k is required. A prior knowledge of the process to be estimated is exploited. In our implementation the pitch detection algorithm, previously described in chapter 3, and a spectrum analysis are both utilised to provide this initial value. This step is found to be adequate for providing the required value. An initial estimate of the error covariance matrix $\hat{P}_k$ associated with $\hat{\mathbf{x}}_k$ is also required. In a statistical sense, this matrix gives the range over which the variates vary from their initial estimates [12]. Therefore in the case of no prior knowledge of the state, a large value for $\hat{P}_k$ is used.

In [37], Poli et al. suggested a new method for estimating the initial values $\hat{\mathbf{x}}_k$ and $\hat{P}_k$. This is called the bootstrapping technique. In this technique the first segment of the test signal (say, the first 100 ms) is time-reversed and fed into the filter. Hence it provides the proper initial values for the resorted original signal. However we have not used this technique in our implementation.

Another parameter that has to be handelled properly is the state covariance

matrix Q. This matrix represents the error in modelling as well as a real driving force. Thus a zero value for Q means that the state variable is a random constant rather than a random walk. It also indicates that the mathematical model is identical to the process in hand. Therefore a non-zero value for Q is used in our implementation. We expect Q to have a small value in high signal-to-noise ratio (SNR) segments of the guitar signal (the steady-state part) whereas it has a large value for the low SNR segments (the attack part of the signal). The value of Q used in our implementation is experimentally chosen, see section (4.6). Computation of both P and Q is explained further in [5].

There are two more parameters that can be adapted to the problem in hand. These are the total number of partials K to be estimated and the sampling interval Δt. The acoustic guitar signal has an insignificant amount of energy at frequencies higher than 3 to 4 KHz (disregarding the attack part) [59]. Consequently we are not trying to model partials higher than this limit. Reducing the total number of partials K is also motivated by the fact that the computational complexity of the algorithm is directly proportional to K.

Generally musical signals are sampled at sampling rate of $f_s = 44.1$ KHz. This is to ensure that all the perceptual audio information is retained in the sample. According to the nature of the acoustic guitar signal that is previously described, the time step parameter defined as $\Delta t = 1/f_s$ can be changed to $\Delta t = D/f_s$ where D is the number of samples to skip (time increment in samples). The effect of using a different sampling interval on the estimated values as well as the overall performance of the proposed algorithm is also investigated.

Another technique that is proven to be useful is adjusting the estimated values [63]. That is, if any of the variates is known to be limited to a certain interval, we adjust the estimated values to fall within this interval.

It is worth mentioning here that since the measurement (the sampled guitar signal) is scalar, the matrix inversion included in computing the Kalman gain

vector (4.19) is reduced to just scalar division.

4.5.2 Smoothing

Smoothing is another technique that can be performed to improve the estimated values of the EKF algorithm. In our case only a fixed-interval smoothing can be applied [5]. Another approach to smoothing is using two EKF algorithms properly combined together. One algorithm performs the forward filtering (the normal one) and the other performs the backward filtering [5], [37]. However in our implementation we did not apply smoothing of any type as the results were quite satisfactory but it remains as a possible improvement.

4.6 Simulation Results

In order to evaluate the performance of the proposed algorithm, it has been implemented for different synthesized signals with known state variables. Firstly we tested some noise-free signals. Then different values of noise have been added to the synthesized signals to check the robustness of the algorithm to noise. In almost all cases the algorithm is proven accurate in estimating the state variables and robust to noise. This step was found necessary also to gain insight on the values of some of the parameters of the EKF algorithm. It also gave us confidence in choosing the appropriate values for the algorithm to start.

We started the EKF algorithm with the initial values given in table (4.2). We started the analysis at time instance $t = 0$. Therefore the initial amplitude values of all partials are set equal to zeros as well as the instantaneous phase. The estimated frequencies from the spectrum analysis are used as the initial frequency values of the different partials. The process noise covariance matrix has been set to 0.001, 0, and 0.1 in its main diagonal for the amplitude, phase and frequency of each partial respectively. The amplitude trajectories of some partials using

Parameter	Initial value
Number of partials K	14
Measurement variance R	0.1
Time increment D	10

Table 4.2: Initial values for the EKF algorithm

the proposed technique and the previous initial values are shown in figure (4.8). We noticed that the second partial was too small to be tracked as well as the 11^{th} and the 12^{th} partials. It is also clear that the results are comparable to those obtained by the STFT-based analysis. The overall decay of the tracked partials is nearly the same. The effect of using different Q values have been also tested. Our results show that the process covariance matrix Q is a critical parameter and has to be carefully chosen. A prior knowledge of the signal to be analyzed and how the variates vary with time can be exploited. For instance if the state variable is known to time-vary slowly, then a small value for Q is used. The estimated fundamental frequency for different Q is shown in figure (4.9). The algorithm is also tested for different values of D. It is obvious that the maximum value of the time increment is limited by the sampling theorem. Using different values for D has no direct effect on the ability of the filter to converge as shown in figure (4.10). Nevertheless it has a direct effect on the estimated run time of the algorithm. The maximum value of D also shows a faster convergence of the filter and the best performance. Table (4.3) shows the estimated run time of the proposed algorithm (measured in seconds) for different four values of D over the same number of measurements (10000 samples) of the guitar test signal. From this table it is clear that the run time is directly related to the time increment. We also noticed that the higher partials are too small to be tracked. This is also due to the fact that higher frequencies decay faster than the lower ones. Therefore the total number of partials can be reduced to 12

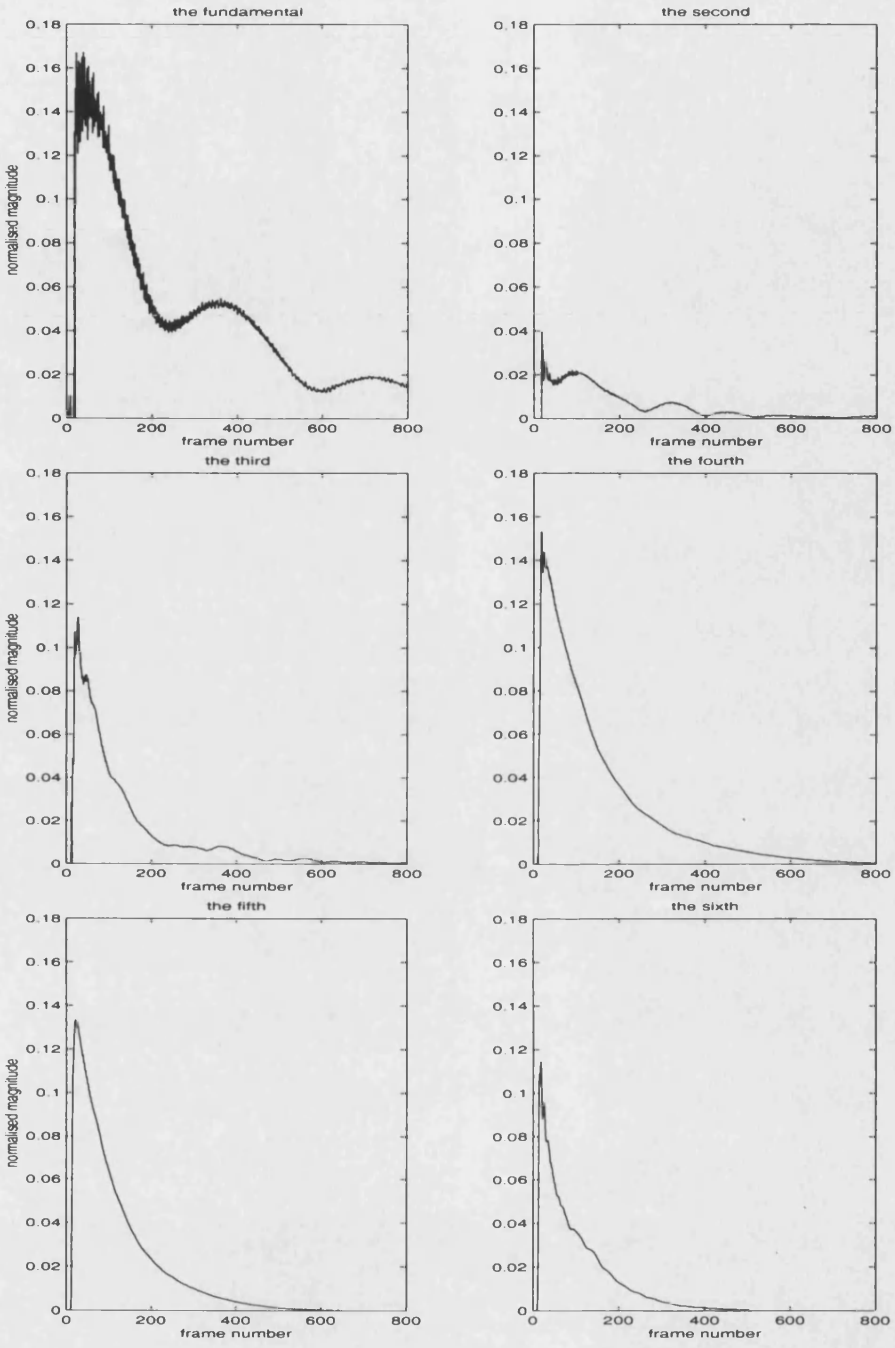

Figure 4.7: Amplitude trajectories of different partials of the guitar signal

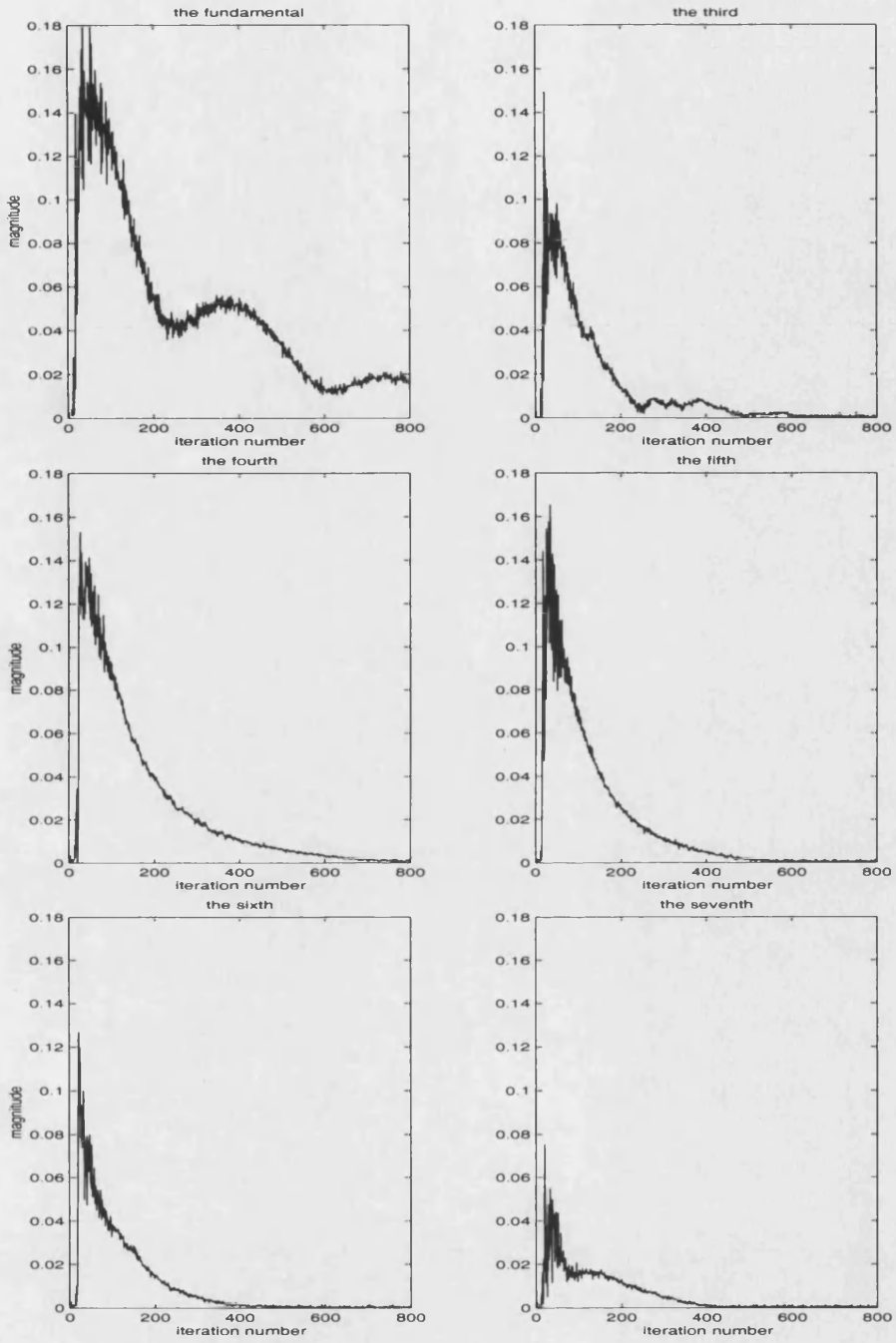

Figure 4.8: Amplitude trajectories of the guitar signal using the EKF algorithm

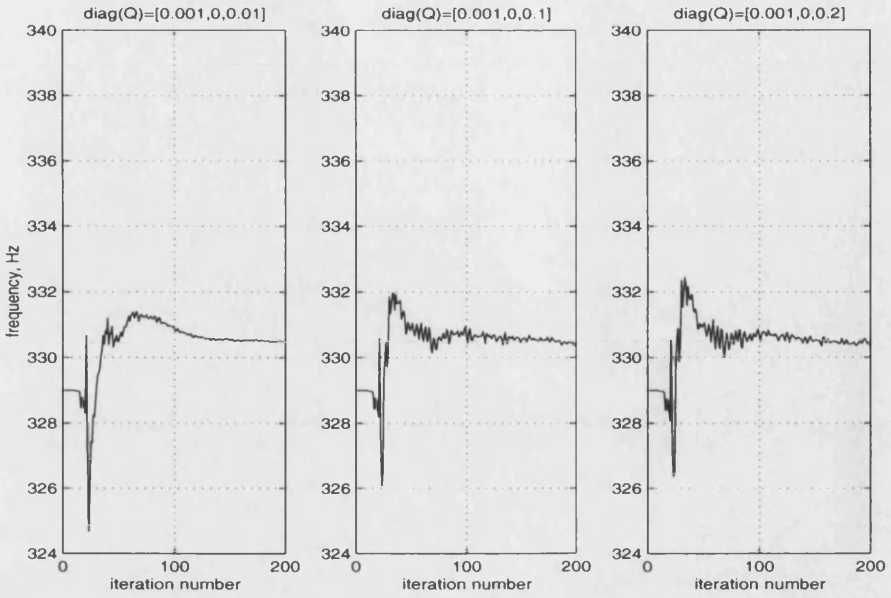

Figure 4.9: The estimated fundamental frequency for different Q values

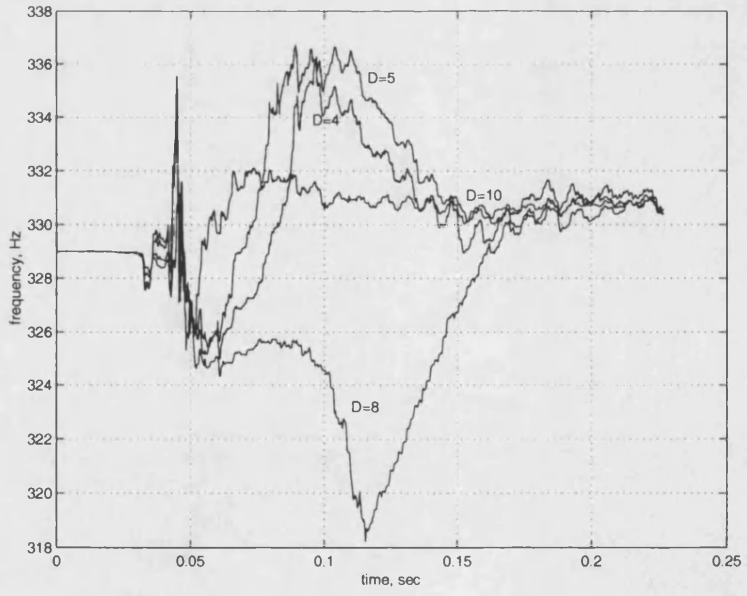

Figure 4.10: The estimated fundamental frequency for different D values

Time increment D	Number of iteration	Estimated time (sec)
D=10	1000	5.7977
D=8	1250	8.0623
D=5	2000	17.4566
D=4	2500	25.4711

Table 4.3: Estimated run time for different values of D.

partials instead of 14. As previously mentioned, the algorithm is faster for lower number of partials which is another motivation for reducing the total number of partials to be estimated. It is worth mentioning at this point that the two lowest body resonances are included in the algorithm.

4.7 Summary

This chapter started by introducing one of the most often used analysis/synthesis techniques, this is the Short-Time Fourier Transform algorithm. Various techniques and algorithms have been utilised for partials tracking of guitar signals using this algorithm. Some of the main disadvantages of the STFT-based technique are also presented in this chapter. The main objective of this chapter, however, is introducing a new algorithm for partials tracking in guitar signals. The proposed new algorithm is based on the theory of the Extended Kalman Filter. Choosing the right model to better suit the problem in hand is not a trivial task unless some prior knowledge of the signal is exploited. Therefore different models have been tested for this purpose. Although the initial model was an obvious choice, the results were not satisfactory. The major problems associated with the initial model have been taken into account in the final model. In this model the guitar signal is modelled as the sum of a sinusoidal model and a statistical model. Each partial in the sinusoidal model is fully described by three

state variables, namely, the amplitude, the frequency and the phase. Both the amplitude and the frequency are modelled as random walks in order to take into account their time-varying nature.

A pre-analysis step is utilised to provide the initial values for the filter to start. This step includes both the pitch detection algorithm and the spectrum analysis of the guitar test signal. This chapter also presents the details of the implementation aspects of the proposed algorithm along with some techniques to optimize the filter performance. Finally the experimental results are presented. These results are compared to the results obtained by the conventional STFT-based analysis for validation and justification. It is worth noting that this tracking method could be adapted to the SMS modelling system [45], with the potential for significant improvement.

Chapter 5

Guitar Model Parameters Estimation

5.1 Introduction

In this chapter the parameter estimation process of the underlying guitar model is described. This model is fully determined by the string model $S(z)$, the two body resonators $B_1(z)$, $B_2(z)$ and the excitation signals $x_{exc}(n)$, $x_{b_1}(n)$ and $x_{b_2}(n)$ as previously described in chapter 2. For the estimation process of the model parameters we follow the same techniques proposed by Tolonen in [55].

Our goal in this chapter is to show that the EKF-based technique, described in the previous chapter, can be applied successfully for the analysis/synthesis process of the guitar signals. Therefore each step in the estimation process is explained using the STFT-based analysis and the EKF-based analysis. In addition a comparative study between the synthesized signal in both cases and the original signal is presented. Furthermore we try to assess the accuracy of the EKF-based algorithm. This is achieved by performing some statistical error-analysis tests on the residual (noise) signal obtained by the proposed algorithm.

This chapter is organised as follows: section 2 is devoted to the estimation of the string model parameters. This involves the estimation of the delay line length L and the loop filter parameters a, g. Thereafter the design of both the loop filter and the fractional delay filter is described. The computation of the sinusoidal model and the residual model is presented in section 3. Section 4 describes the design of the body resonators which is accomplished by estimating each resonator's centre frequency and bandwidth. The computation of the excitation signals of the string model and the two body resonators is presented in section 5. In section 6 a comparison between the synthesized signal in both cases and the original one is presented. Some statistical tests for the analysis of the EKF-residual signal are also given in this section. Finally the chapter summary is given in section 7.

5.2 Estimation of the String Model Parameters

Recall from chapter 2 that the string model of the underlying guitar model involves three main parts, namely, the loop filter $H(z)$, the delay line L and the fractional delay filter $F(z)$. In the following subsections the estimation process of each part is further explained.

5.2.1 Loop Filter Design

The purpose of this section is to estimate the loop filter parameters a and g as well as the design of this filter. Recall that the loop filter is the main lossy part in the string model when an all-pass filter is used for fine tuning the pitch period. Therefore it determines the frequency-dependent damping of the synthetic signal. The aim of the analysis process, previously described in chapter 4, is to provide the amplitude trajectory of each partial of the real guitar signal. The damping factors of such trajectories determine the magnitude responses of the loop filter [59]. Once the prototype frequency response of the loop filter is computed, the loop filter is designed in a weighted least-square sense [55].

The design of the loop filter is organized as follows: first the decay rate of each amplitude trajectory is computed. Hence the frequency-dependent magnitude response of the loop filter is evaluated. A method for the loop filter design is then described.

5.2.1.1 Computation of the Decay Rates.

In this section the decay rate of each detected partial of the guitar signal is evaluated. In the ideal case, the amplitude trajectory should decay exponentially in the steady-state part of the signal. This situation is hardly met in practice due to non-linearities which result in fluctuations as previously shown in chapter 4. Smoother trajectories are obtained by integrating them backward in

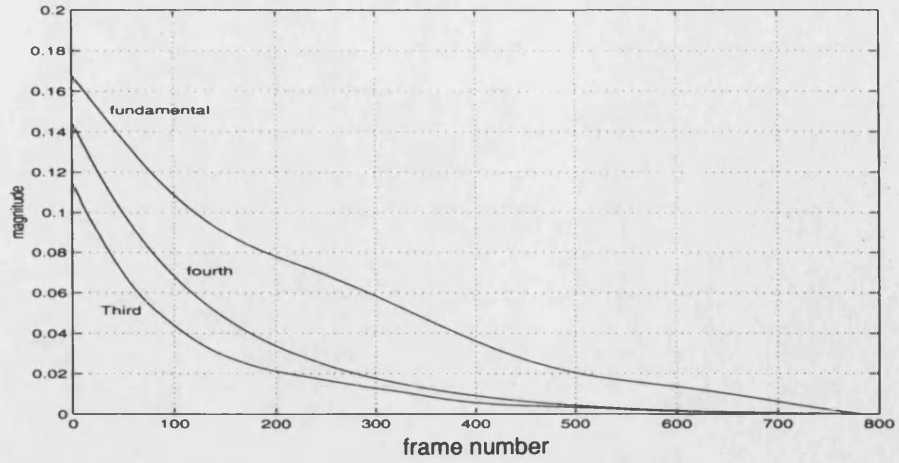

Figure 5.1: Schroeder-integrated curves of some partials in the STFT case

time [55]. This technique was first used by Schroeder for reverberation time measurements [44]. In [19] Jot generalised this idea in the frequency domain to yield the energy decay relief curves. This is the backward integration of the STFT of a time-domain signal. Hence the decay rates of the partials at a discrete set of frequencies are computed. For each amplitude trajectory, the energy decay relief curves $A_r(l)$ are defined as [55]:

$$A_r(l) = \sum_{m=l}^{\infty} E_r(m) \tag{5.1}$$

where $E_r(l)$ is the amplitude trajectory of the r^{th} partial evaluated at frame number l and $A_r(l)$ is the corresponding energy decay relief curve. Some of the computed curves are shown in figures (5.1) and (5.2) in the case of the STFT analysis and the EKF analysis respectively. It is clear that these curves are smoother with less fluctuations than the original ones. Therefore these curves are more appropriate for linear regression. For comparison purposes the curves are evaluated over the same number of samples in the STFT-analysis and the EKF-analysis.

It is important to clarify that the energy decay relief curves preserve only the decay rates of the original curves but they do not retain the original amplitude

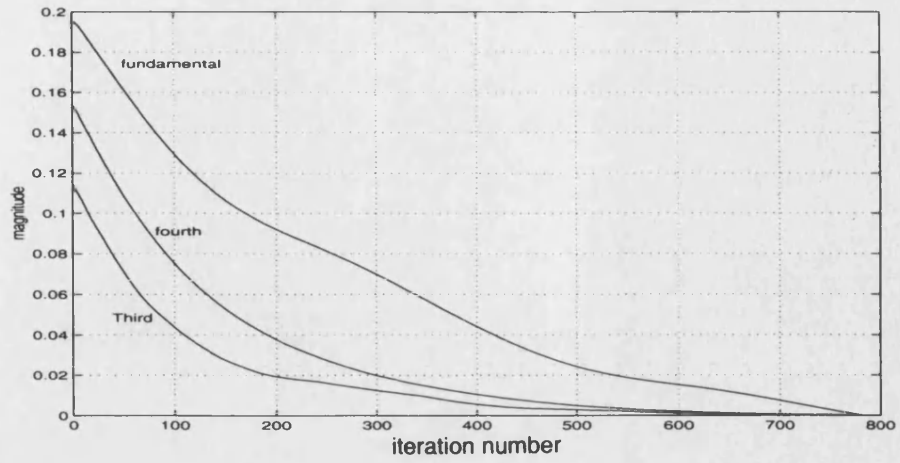

Figure 5.2: Schroeder-integrated curves of some partials in the EKF case.

levels. Therefore only the decay rates of the partials are obtained from these curves. Note that there is a difference in the sampling interval used in the STFT-analysis and the EKF-analysis. Hence we expect the curves in both cases to be slightly different. Nevertheless it is clear from figures (5.1) and (5.2) that the curves have nearly the same decay rates.

5.2.1.2 Magnitude Responses of the Loop Filter

On a decibel scale each trajectory should be a straight line. A segment of every Schroeder-integrated curve is chosen to fit a straight line to it. The slope B_k of each partial is evaluated. The corresponding magnitude response of the loop filter is computed as follows [59]:

$$g_k = 10^{\frac{B_k * L_1}{20 * H}} \tag{5.2}$$

where L_1 is the total delay estimated from chapter 3, H is the hop size used in the STFT algorithm or the increment parameter D used in the EKF-algorithm and g_k is the magnitude response of the loop filter corresponding to the k^{th} partial.

Table (5.1) gives the estimated slopes of some partials in both cases of the STFT algorithm and the EKF algorithm.

Partial number	STFT	EKF
first	-0.2306e-03	-0.2396e-03
third	-0.4913e-03	-0.5405e-03
fourth	-0.4629e-03	-0.4659e-03
fifth	-0.6138e-03	-0.6048e-03

Table 5.1: Slopes of various partials in the STFT and the EKF cases.

5.2.1.3 Computation of the Loop Filter Parameters

The values g_k determine the desired magnitude responses of the loop filter at frequencies ω_k of the k^{th} partial. That is the prototype magnitude response of the loop filter. The purpose of this section is to design a loop filter that matches the overall decay rate of the guitar signal rather than the individual partial decay rate. We follow here the same design method previously described by Tolonen [55]. A weighted-least square method is used for determining the loop filter parameters. The function to be minimized is given by:

$$E = \sum_{k=0}^{K-1} W(k) \left[G_l(\omega) - g_k \right]^2 \tag{5.3}$$

where K is the number of partials incorporated in the filter design, $W(k)$ is an appropriate non-negative weighting function and $G_l(\omega)$ is the loop filter magnitude response given by equation (2.17). The weighting function has to be chosen in a way such that partials of longer decay are weighted heavily than the ones with shorter decay. In [24] Valimaki et al. suggest the use of a value given by

$$W(k) = \frac{1}{1 - g_k} \tag{5.4}$$

The error function E is minimized with respect to both a and g where

$$0 < g < 1,$$

$$-1 < a < 0$$

Parameter	STFT	EKF
a	-0.0225	-0.0202
g	0.9994	0.9993

Table 5.2: Loop filter parameters

This is achieved by using optimization tools, e.g. MATLAB. For the small partials such as the second and the higher ones, the weighting function is set equal to zero.

At this point it is worth mentioning that we are trying to match the magnitude response of the loop filter rather than the phase response. This is due to the fact that the dispersion effect of the one-pole filter is too small to affect the result [24]. The estimated values of both a and g in the STFT algorithm and the EKF algorithm are shown in table (5.2).

5.2.2 Delay Line Estimation

Recall from chapter 3 that the total delay of the string model is the real valued number L_1 defined by [59]

$$L_1 = \frac{f_s}{f_0} \tag{5.5}$$

where f_s is the sampling rate and f_0 is the estimated pitch period. This delay can be further expressed as

$$L_1 = L + D_l + D_f \tag{5.6}$$

where

L is the delay-line length which is an integer value,

D_l is the phase delay of the loop filter, and

D_f is the phase delay of the fractional delay filter.

Once the loop filter is designed, the parameters a and g are determined. Accordingly the phase delay D_l of the loop filter can be estimated. It is given by

$$\begin{aligned} D_l &= -\frac{\theta_l(\omega)}{\omega} \\ &= -\frac{1}{\omega}\tan^{-1}\left(\frac{a\sin(\omega)}{1+a\cos(\omega)}\right) \end{aligned} \tag{5.7}$$

where $\theta_l(\omega)$ is the loop filter phase response given by equation (2.18). Equation (5.7) gives the exact phase delay of the loop filter. It can be approximated for low frequencies by expanding the arctan function and ignoring the higher order terms. This yields

$$D_l = \frac{-a}{1+a} \tag{5.8}$$

Hence the delay-line length evaluates to

$$L = floor(L_1 - D_l) \tag{5.9}$$

where $floor(x)$ gives the nearest integer less than or equal to the real-valued argument x.

5.2.3 Fractional Delay Filter Design

The fractional delay filter is fully determined by its parameter c. In order to compute c, the phase delay of the fractional delay filter needs to be estimated. Recall from chapter 2 that the phase response of the fractional delay filter is given by

$$\theta_f(\omega) = \tan^{-1}\left(\frac{-\sin(\omega)}{c+\cos(\omega)}\right) - \tan^{-1}\left(\frac{-c\sin(\omega)}{1+c\cos(\omega)}\right) \tag{5.10}$$

Expanding the arctan function and ignoring terms of higher orders yields

$$\theta_f(\omega) = \frac{-\sin(\omega)}{c+\cos(\omega)} + \frac{c\sin(\omega)}{1+c\cos(\omega)} \tag{5.11}$$

which can be further approximated for low frequencies to

$$\begin{aligned} \theta_f(\omega) &= \frac{-\omega}{c+1} + \frac{c\omega}{1+c} \\ &= \omega\left(\frac{c-1}{1+c}\right) \end{aligned} \tag{5.12}$$

Therefore the phase delay of such filter evaluates to

$$\begin{aligned} D_f &= -\frac{\theta_f(\omega)}{\omega} \\ &= \frac{1-c}{1+c} \end{aligned} \tag{5.13}$$

where D_f is determined by

$$D_f = (L_1 - D_l) - L \tag{5.14}$$

Thus the parameter c is computed as

$$c = \frac{1-D_f}{1+D_f} \tag{5.15}$$

5.3 Sinusoidal and Residual Models

The purpose of this section is to compute the sinusoidal model that corresponds to the harmonic part of the guitar test signal as well as the residual model. The analysis process, previously described in chapter 4, describes the significant partials present in the guitar signal in terms of their amplitude, phase, and frequency. To compute the sinusoidal model in the STFT-based analysis, the instantaneous amplitude $E_k(m)$ and the instantaneous phase $\theta_k(m)$ corresponding to the k^{th} partial need to be evaluated.

The output of the STFT-based analysis is a set of frames. Each partial k has three values, namely, amplitude $A_k(l)$, frequency $\omega_k(l)$ and phase $\varphi_k(l)$ at different frames l. These values are separated in time by the hop size parameter H. The instantaneous amplitude $E_k(m)$ at time instance m is obtained by linearly

interpolating the amplitude values between frames according to

$$E_k(m) = A_k(l-1) + \Big(\frac{A_k(l) - A_k(l-1)}{H}\Big)\, m \tag{5.16}$$

for $m = 0, 1, \ldots, H-1$.

To compute the instantaneous phase $\theta_k(m)$, cubic interpolation between frames is utilised using the four available values $\omega_k(l-1)$, $\omega_k(l)$, $\varphi_k(l-1)$ and $\varphi_k(l)$. Cubic interpolation is given in more detail in [31].

The cubic function used in the interpolation is given by

$$\theta_k(m) = \xi + \gamma\, m + \eta\, m^2 + \zeta\, m^3 \tag{5.17}$$

where ξ, γ, η and ζ are the polynomial coefficients that need to be determined. The solution of this polynomial evaluates to [31]

$$\theta_k(m) = \varphi_k(l-1) + \omega_k(l-1)\, m + \eta\, m^2 + \zeta\, m^3 \tag{5.18}$$

where

$$\eta = \frac{3}{H^2}\mathbf{I} - \frac{1}{H}\mathbf{II} \tag{5.19}$$

$$\zeta = \frac{-2}{H^3}\mathbf{I} - \frac{1}{H^2}\mathbf{II} \tag{5.20}$$

I and **II** are given by

$$\mathbf{I} = \varphi_k(l) - \varphi_k(l-1) - \omega_k(l-1)\, H + 2\pi\, M$$

$$\mathbf{II} = \omega_k(l) - \omega_k(l-1)$$

Both η and ζ are functions of the parameter M which has to be chosen in such a way to provide maximally smooth instantaneous phase [31]. This is obtained by having M equals to

$$M = round(q) \tag{5.21}$$

where

$$q = \frac{1}{2\pi}\Big[\varphi_k(l-1) + H\omega_k(l-1) - \varphi_k(l) + \frac{H}{2}\big(\omega_k(l) - \omega_k(l-1)\big)\Big] \tag{5.22}$$

The $round(x)$ defines the integer value that is closest to the argument x.

Note that this value of q is the root of the second derivative of $\theta_k(m)$. Choosing M closest to q ensures that a maximally smooth instantaneous phase is obtained [31].

By applying equation (5.18) at all frame boundaries, the phase envelope $\theta_k(m)$ is evaluated. Accordingly the sinusoidal model $x_{sin}(m)$ is computed using additive synthesis. That is

$$x_{sin}(m) = \sum_{k=1}^{k=N} E_k(m) \cos(\theta_k(m)) \tag{5.23}$$

The value of N in this sum means the number of partials present in the signal at time instance m. Hence the residual model $x_{res}(n)$ is computed using subtractive synthesis as follows

$$x_{res}(m) = x(m) - x_{sin}(m) \tag{5.24}$$

where $x(m)$ is the original guitar test signal. Figure (5.3) shows the real guitar test signal, the sinusoidal signal x_{sin} and the residual signal x_{res} obtained using the STFT-based analysis. Note that the residual signal still includes the lowest body resonances.

In the EKF-based analysis, the algorithm provides the amplitude and phase trajectories at time instances separated by the sampling parameter D. Therefore the instantaneous amplitude and phase are obtained by linear interpolation as previously described in the STFT-based analysis. Figure (5.3) shows the sinusoidal and residual signals obtained by the EKF-based algorithm.

Although the lowest body resonances have been extracted already by the EKF algorithm, we did not remove them from the residual signal. This is for comparison reasons with the STFT-residual signal which still includes the body resonances.

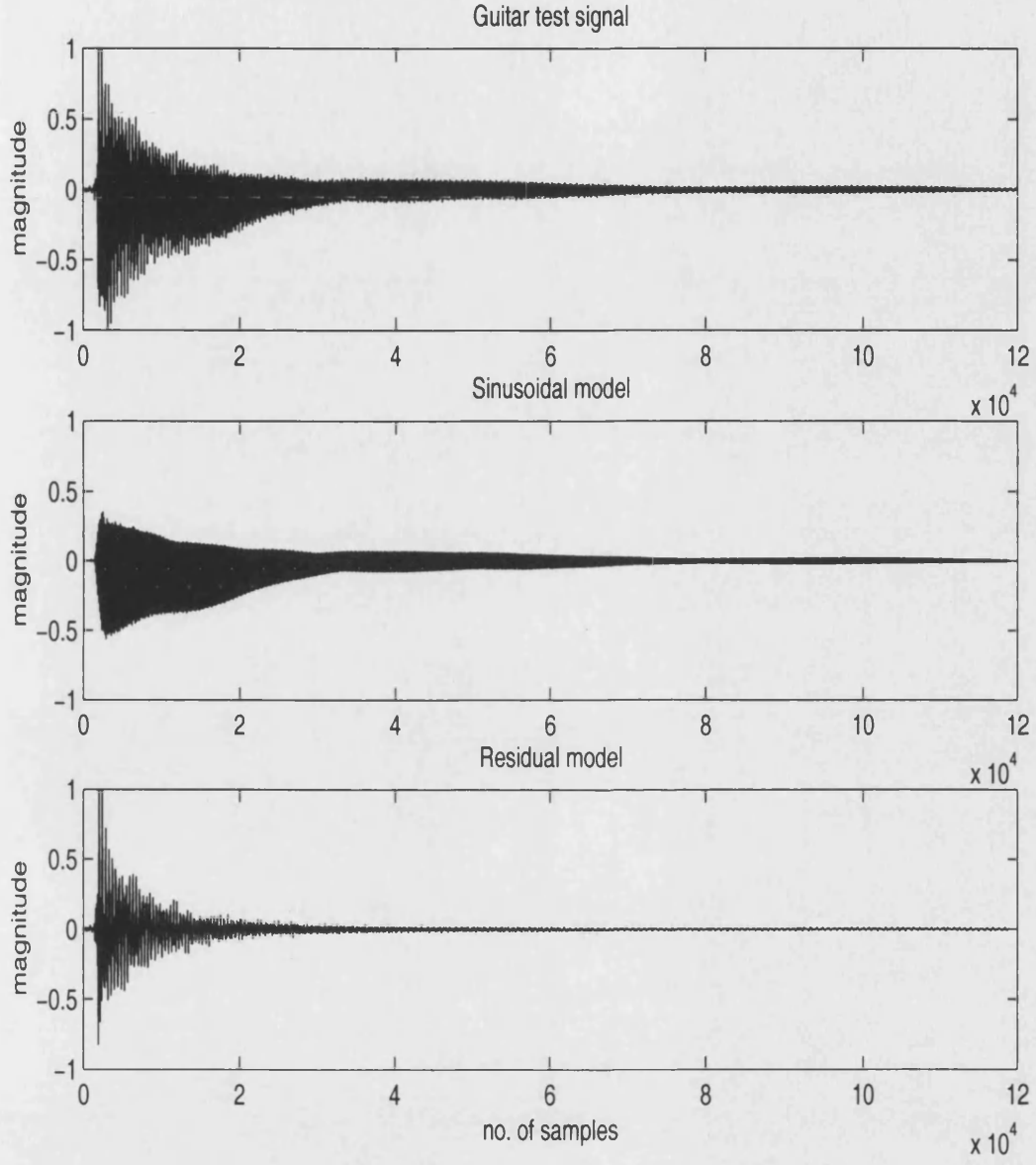

Figure 5.3: Sinusoidal and residual models using the STFT

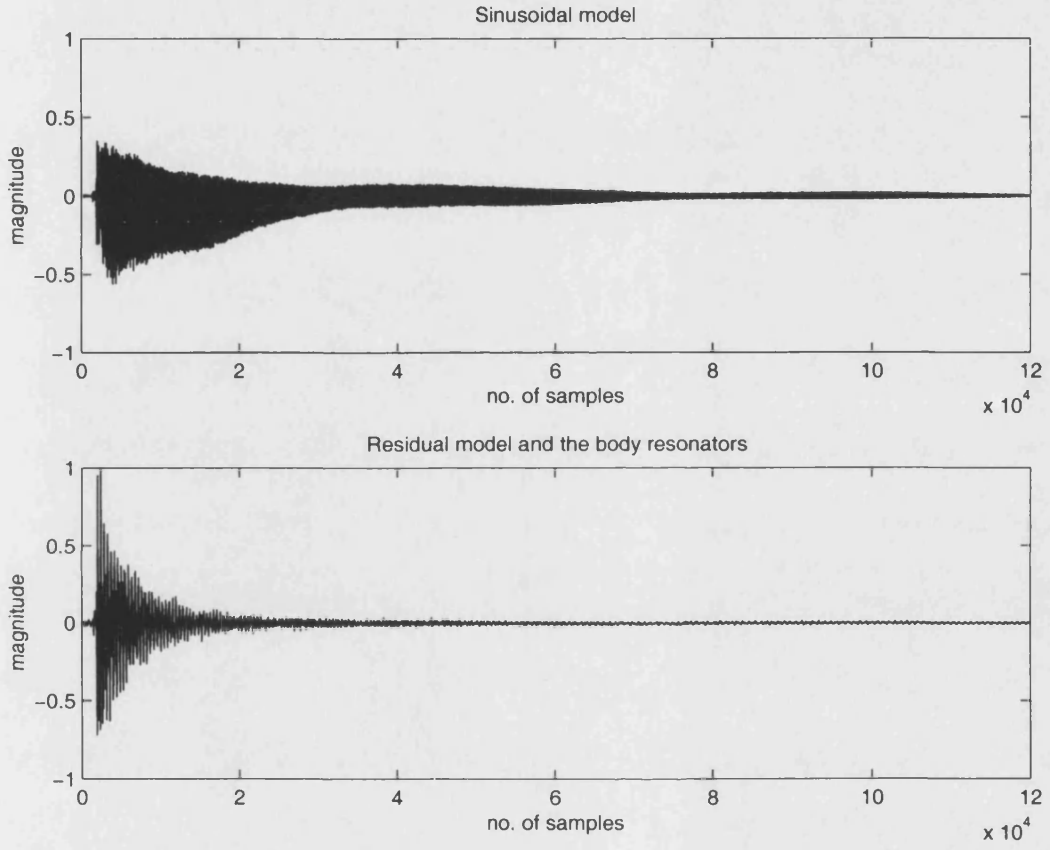

Figure 5.4: Sinusoidal and residual models using the EKF

5.4 Estimation of the Body Resonators Parameters

In this section we present the estimation process of the body resonators $B_1(z)$ and $B_2(z)$. Recall from chapter 3 that each resonator is fully determined by two parameters: the centre frequency ω_0 and the bandwidth $\Delta\omega$. This section starts by describing the estimation of those parameters in the STFT-based analysis case followed by the estimation process in the EKF-based analysis.

In the STFT-based analysis the residual signal $x_{res}(m)$ contains information about the plucking event, the body response and the attack transient. To estimate the body resonator parameters, the STFT of the residual signal is computed. In this case the window size is adapted to four times the lowest body resonance. The centre frequencies ω_{01} and ω_{02}, corresponding to $B_1(z)$ and $B_2(z)$ respectively, is readily available from the STFT.

The amplitude trajectories of the lowest body resonances $E_{b_1}(l)$, $E_{b_2}(l)$ are obtained by utilising the peak detection and the peak continuation algorithms previously described in chapter 4. Figure (5.4) shows a 3D-plot of the STFT of the residual signal whereas $E_{b_1}(l)$ and $E_{b_2}(l)$ are shown in figure (5.4). Smoother trajectories are obtained by using the EDR curves as described in section (5.1). A straight line is fitted to each EDR curve on a dB-scale and its slope is computed. The radius of the resonator pole is then obtained by [55]

$$r_j = 10^{s_j/(20*H_l)} \qquad \text{for} \quad j = 1, 2 \tag{5.25}$$

where s_j is the slope of the j^{th} body resonator, H_l is the hop size parameter used in computing the STFT or the increment parameter D in the EKF case, and r_j is the corresponding pole radius. Note that H_l is different from H, the hop size used in the previous STFT, since the window size is different. For values of r_j

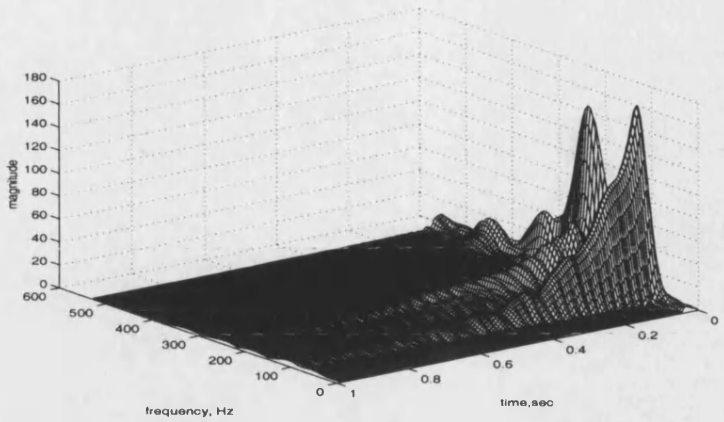

Figure 5.5: 3D-plot of the STFT of the residual signal

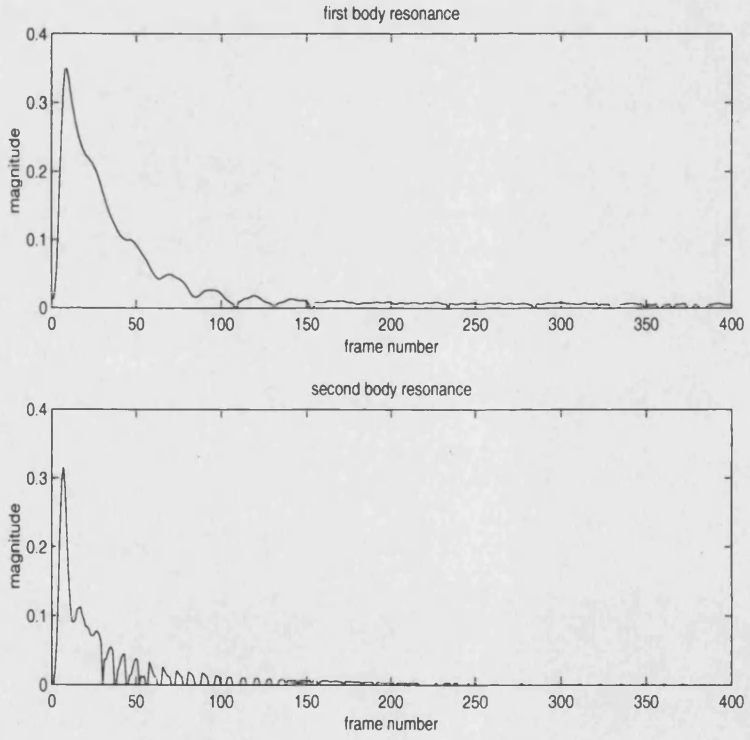

Figure 5.6: Amplitude trajectories of the body resonances using the STFT

Estimated values	$B_1(z)$	$B_2(z)$
Slope	-0.0006	-0.0006
Centre frequency f_0	100.78	212.78
Pole radius r	0.999	0.999
Bandwidth $\Delta\omega$	0.002	0.002

Table 5.3: Body resonators parameters using the STFT

Estimated values	$B_1(z)$	$B_2(z)$
Slope	-0.0009	-0.0004
Centre frequency f_0	102.29	209.01
Pole radius r	0.999	0.999
Bandwidth $\Delta\omega$	0.002	0.002

Table 5.4: Body resonators parameters using the EKF

close to unity, the corresponding bandwidth $\Delta\omega_j$ is evaluated according to [36]

$$\Delta\omega_j \simeq 2(1 - r_j) \tag{5.26}$$

The estimated values of ω_{01}, ω_{02}, r_1, r_2, $\Delta\omega_1$ and $\Delta\omega_2$ using the STFT-based analysis are given in table (5.3). In the EKF-based analysis the two lowest body resonances have been included already in the final model previously described in chapter 4. Therefore there is no need to apply the filter once more for estimating the amplitude trajectories of the body resonances. The estimation process continues by estimating the decay rates of such trajectories and consequently computing the corresponding bandwidth. This step is similar to that described in the STFT-based case. Figure (5.4) shows the amplitude trajectories of the two lowest body resonances. Table (5.4) shows the results of the estimated values using the EKF-based algorithm.

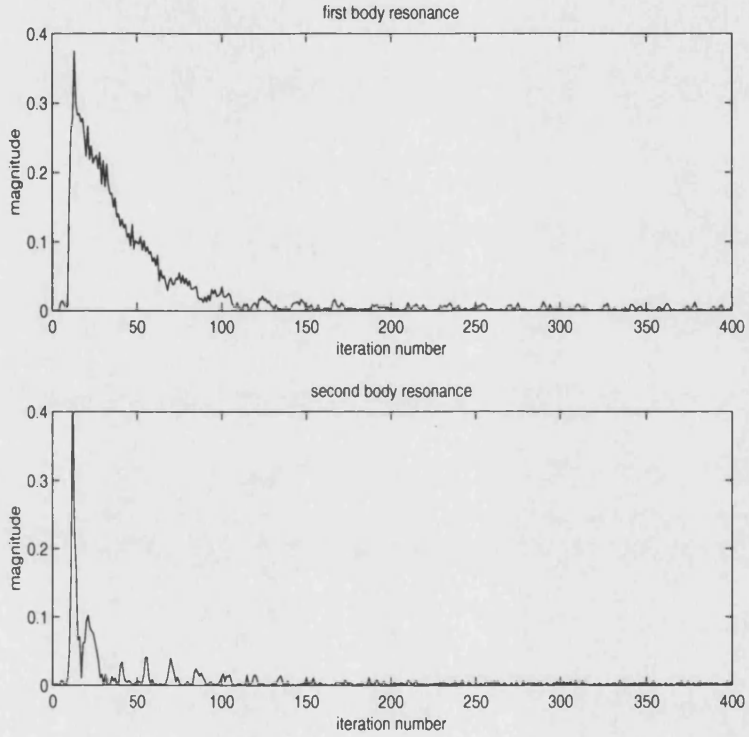

Figure 5.7: Amplitude trajectories of the body resonances using the EKF

5.5 Estimation of the Excitation Signals

In this section, we present the estimation process of the excitation signals x_{exc}, x_{b_1} and x_{b_2} for the string model $S(z)$ and the body resonators $B_1(z)$ and $B_2(z)$ respectively.

5.5.1 String Model Excitation Signal

In [55] Tolonen has proposed a technique for estimating the string model excitation signal. In his technique the residual signal $x_{res}(n)$ is post-processed to be applicable as an excitation signal for the string model. This is achieved by inverse filtering the original guitar signal, expressed as a sum of the sinusoidal model and the residual model, with the reciprocal of the string model $S^{-1}(z)$. That is

$$X_{inv}(z) = S^{-1}(z)\left[X_{res}(z) + X_{sin}(z)\right] \tag{5.27}$$

where $S(z)$ is the string transfer function given by (2.22), $X_{res}(z)$ and $X_{sin}(z)$ are the z-transforms of the residual and the sinusoidal models respectively. In the time-domain, equation (5.27) can be re-written as

$$x_{inv}(n) = x_{rinv}(n) + x_{sinv}(n) \tag{5.28}$$

where $x_{rinv}(n)$ is the inverse-filtered residual signal and $x_{sinv}(n)$ is the inverse-filtered sinusoidal model. The inverse-filtered signal of the sinusoidal signal $x_{sinv}(n)$ contains an estimation of the error produced by the string model [55]. The attack part of this signal is used to equalize the spectral content of the attack part of the residual signal. Therefore the signal $x_{sinv}(n)$ is truncated after the attack part. This is achieved by multiplying this signal by the right half of a window function such as the Hanning window. Hence the excitation signal is given by

$$x_{exc,b} = x_{sinv}(n)W_{half}(n) + x_{rinv}(n) \tag{5.29}$$

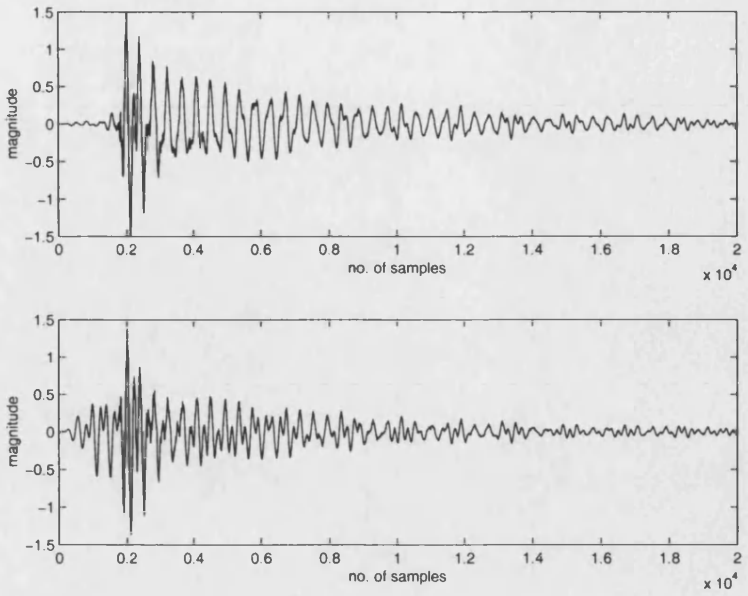

Figure 5.8: String model excitation signal

where $W_{half}(n)$ is a one-sided window function. Note that the excitation signal $x_{exc,b}(n)$ includes the body resonances as well. Therefore the excitation signal $x_{exc}(n)$ is obtained by subtracting the sinusoidal models of the body resonances from the signal $x_{exc,b}$. That is

$$x_{exc}(n) = x_{exc,b}(n) - x_{bsin1}(n) - x_{bsin2}(n) \tag{5.30}$$

where $x_{bsin1}(n)$ and $x_{bsin2}(n)$ are the sinusoidal models of the first and second body resonance respectively. Figure (5.8) shows the excitation signal $x_{exc,b}(n)$ in the top figure whereas the excitation signal $x_{exc}(n)$ is shown in the bottom figure. The attack part of this signal poses one of the major problems with the STFT-based analysis; that is the trade-off between the time resolution and the frequency resolution briefly described in chapter 4. To resolve the lowest body resonance (100Hz), a wider window has been used in the analysis which, in turn, results in a poor time resolution. Therefore the time instance of the peak can not be determined accurately using the STFT technique.

In the EKF-based case, the excitation signal of the string model can be esti-

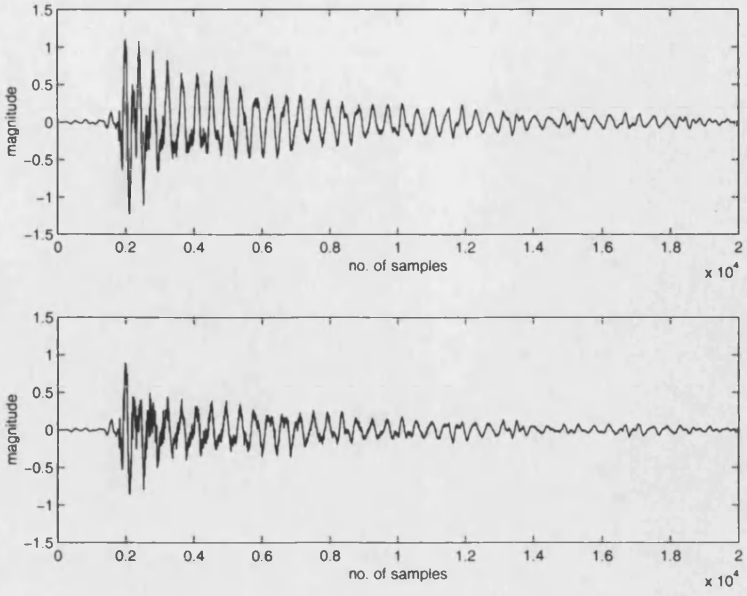

Figure 5.9: String model excitation signal in the EKF-based case

mated by following the same procedure previously used in the STFT-based analysis. The results are shown in figure (5.9) where the excitation signal $x_{exc,b}(n)$ is shown in the top figure and the excitation signal $x_{exc}(n)$ is shown in the bottom figure.

5.5.2 Body Resonators Excitation Signals

Recall that the second-order transfer function of the body resonator is given by

$$B(z) = (1-b)\,\frac{1-z^{-2}}{1-2b\cos(\omega_0)\,z^{-1}+(2b-1)z^{-2}}$$

This filter has no realizable inverse since it has two zeros at $z = 0$ and $z = 1$. To estimate the excitation signal of each body resonance a pseudo-inverse filter is designed instead [55]. This filter can be written as

$$\hat{H}(z) = \frac{b_{f_0}}{1-2b\cos(\omega_0)\,z^{-1}+(2b-1)z^{-2}} \tag{5.31}$$

where

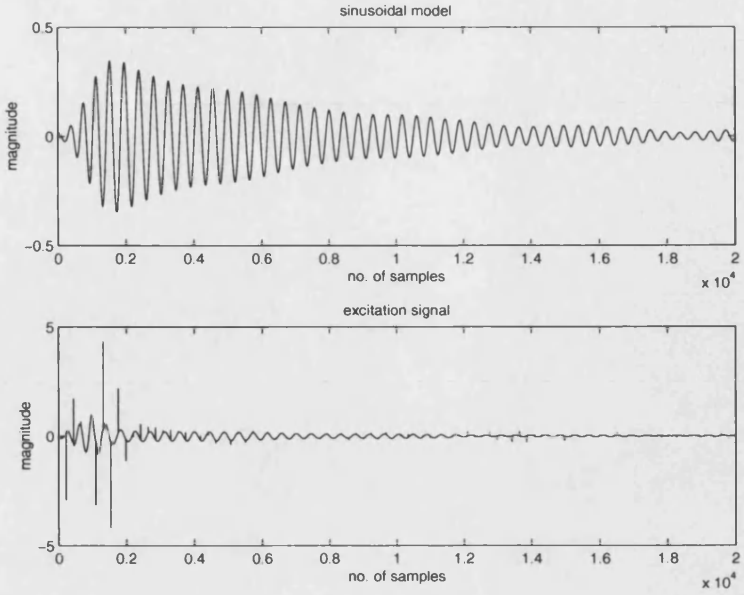

Figure 5.10: Body resonator excitation signal in the STFT case

$b_{f_0} = (1-b)[1-e^{-2j\omega_0}]$,

$\omega_0 = 2\pi\, f_0/f_s$,

$b = 1/(1+\tan(\Delta\omega/2))$,

$\Delta\omega$ is the 3-dB bandwidth of each resonator in radians.

The parameter b_{f_0} gives the value of the numerator of the original body resonators at the centre frequency f_0 of each resonance. Hence the inverse filter $\hat{H}^{-1}(z)$ is given by

$$\hat{H}^{-1}(z) = \frac{1}{b_{f_0}}\left(1 - 2b\cos(\omega_0)\, z^{-1} + (2b-1)\, z^{-2}\right) \tag{5.32}$$

The excitation signals $x_{b1}(n)$ and $x_{b2}(n)$ are then obtained by inverse-filtering the corresponding sinusoidal models $x_{bsin1}(n)$ and $x_{bsin1}(n)$ with the filter $\hat{H}^{-1}(z)$. That is

$$X_{b1}(z) = \hat{H}^{-1}(z)\, X_{bsin1}(z) \tag{5.33}$$

$$X_{b2}(z) = \hat{H}^{-1}(z)\, X_{bsin2}(z) \tag{5.34}$$

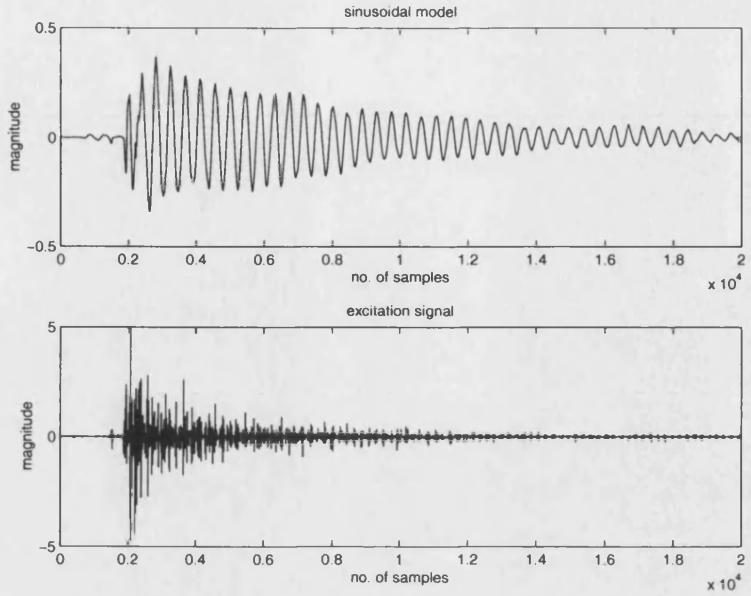

Figure 5.11: Body resonator excitation signal in the EKF case

where $X_{b1}(z)$, $X_{b2}(z)$, $X_{bsin1}(z)$ and $X_{bsin2}(z)$ are the z-transforms of the excitation signals and the sinusoidal models of the lowest two body resonances [55]. Figures (5.10) and (5.11) show the sinusoidal model $x_{bsin1}(n)$ (top) of the lowest body resonance and the excitation signal $x_{b1}(n)$ (bottom) of the body resonator in the STFT and the EKF cases respectively.

5.6 Results and Discussion

In this section an analysis of the re-synthesized signal, obtained by the proposed techniques, is described. The results are compared to those obtained by the STFT-based technique. This section also presents a time-frequency analysis of the residual (noise) signal of the EKF-based technique along with some statistical analysis tests.

5.6.1 Analysis of the Re-synthesized Signal

The re-synthesized signal is obtained by applying the excitation signals previously estimated to the guitar model.

This signal is shown in figure (5.12) where the top graph shows the original guitar signal, the middle graph shows the re-synthetic signal using the STFT-based technique and the bottom graph shows the corresponding signal in the EKF-based technique. It is worth mentioning here that the guitar model is just a linear model that is not capable of producing all non-linearities present in the real guitar signal. However, it is quite adequate for our purpose of study which is the analysis process rather than the modelling process itself. It is clear from the figure that the signals retain the original level of amplitude of the original one as well as the overall decay rate. In addition the re-synthetic signals have quite the same attack part up to the truncation point of the equalized signal.

The original signal and the re-synthesized signals using the STFT and the EKF techniques are appended to the thesis on an audio CD-rom.

5.6.2 Analysis of the Residual Signal

In this section we analyse the residual signal in order to have some insight on the performance of the proposed techniques. This is achieved by studying the residual signal properties in the time-domain and in the frequency-domain. Further the average value and the variance of the residual are computed. Generally the smaller the residual the more accurate is the estimation process.

Figure (5.13) shows the residual signals of the STFT and the EKF cases. A closer look at the attack part of the two signals is shown in figure (5.14) which imposes the time-resolution problem of the STFT-based technique. Some basic error-analysis tests have also performed on the residual signal. Table (5.5) shows the estimated values of the mean, the variance, and the standard deviation of this signal.

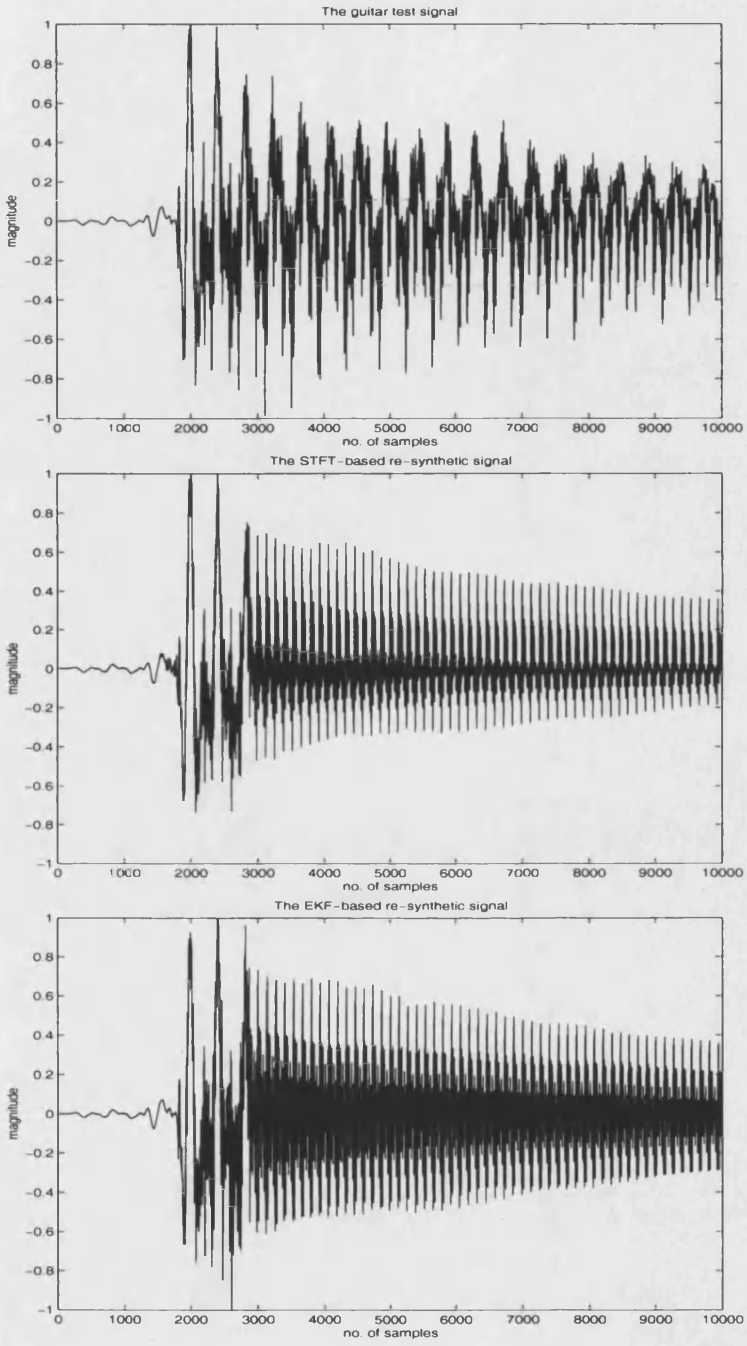

Figure 5.12: The re-synthetic guitar signal

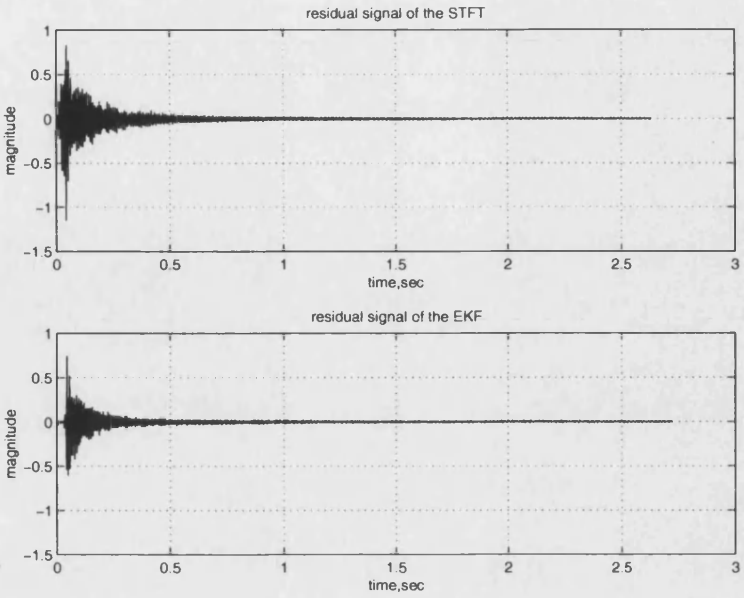

Figure 5.13: The time-domain residual signal

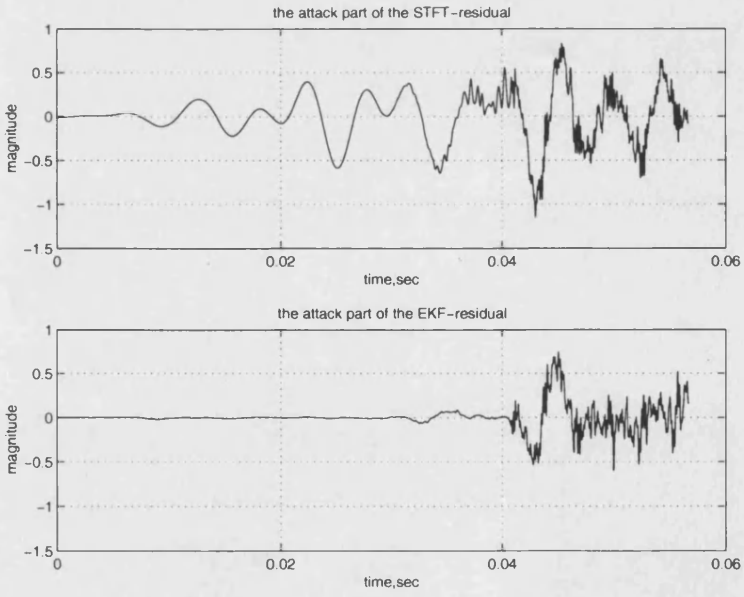

Figure 5.14: The attack part of the residual signal

Criteria	The EKF-residual	The STFT-residual
Average value	0.00083	0.0012
Variance	0.00082	0.0026
Standard deviation	0.0287	0.0512

Table 5.5: Statistical values of the residual signals

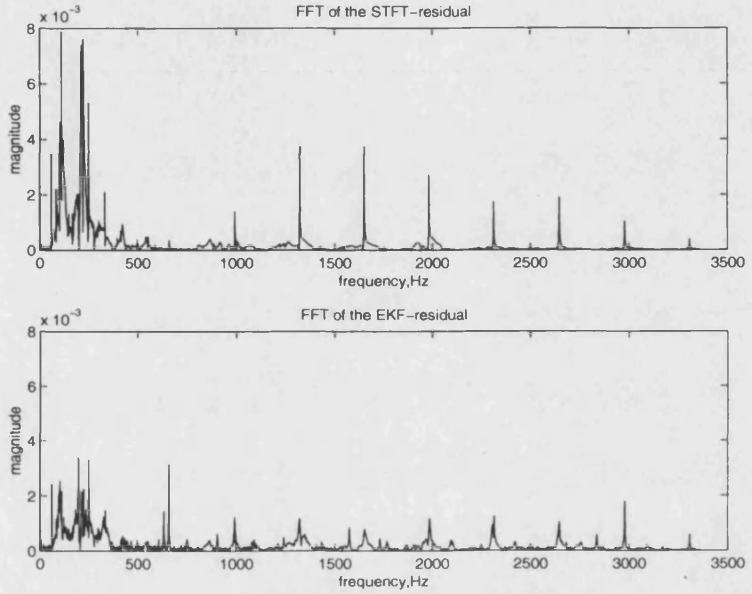

Figure 5.15: FFT of the residual signal

From figure (5.13) it is clear that the EKF-residual signal is smaller than the STFT-residual signal. Comparing the estimated values in table (5.5) implies that the EKF-based technique is more efficient than the STFT-based analysis.

In the frequency-domain the spectrum of the residual signal is shown in figure (5.15). This figure shows that the spectrum of the STFT-residual signal has more partials with even higher magnitudes than the corresponding ones in the EKF-based technique.

5.6.3 Discussion

Throughout this chapter the proposed EKF-based technique has proven to be efficient and accurate. This indicates that the proposed technique can be used not only for tracking but also for the calibration process. Nevertheless we believe that the calibration process is an open area for further improvement.

5.7 Summary

This chapter has presented the calibration process of the underlying guitar model. The estimation process is carried out in the STFT-based analysis as well as the EKF-based analysis. Each step in the estimation process is explained in details along with the main differences between the two techniques. This chapter has also presented an analysis of the re-synthetic signal and the residual signal.

Chapter 6

Conclusions and Future Work

Starting with a physically-relevant model of the acoustic guitar which is often described and used in the literature, some new tools for the analysis-based synthesis of the guitar signals have been presented in this study. The new tools are the pitch detection algorithm and the partials tracking algorithm. The main results of the new techniques are presented in this chapter and the advantages and disadvantages of the proposed techniques are described. Finally, ideas for future work and some suggestions for improvements in both the modelling process and the analysis process are presented.

6.1 Thesis Summary

Physical modelling of a real musical instrument involves two major steps. The first step is the design of a relevant model that simulates closely the real instrument. This is achieved by studying the physical structure of the instrument along with the mathematical equations that govern its behaviour. Second the physical model must be calibrated to a certain instrument which is accomplished by the analysis of real signals of this particular instrument. In our study we have focused on the second step.

After introducing a general overview of the various techniques of physical modelling, the construction of the instrument under study; the acoustic guitar, is briefly described in chapter 1.

The design of a physically relevant model of the acoustic guitar is presented in chapter 2. In this chapter the principles of the digital waveguide technique are reviewed and the formulation of a single delay line model is illustrated. We also present the KS-algorithm for simulating plucked strings and drum-like tones. Based on the extensions of this algorithm and the principle of commuted waveguide, a guitar model is designed. Other non-linearities that can be incorporated in the guitar model are briefly presented.

Having designed a model of the acoustic guitar, the next step is the analysis of real guitar signals in order to calibrate the model. This analysis process starts by estimating the pitch period of the guitar signal which in turn determines the total delay of the string model. For this purpose a novel method has been presented in chapter 3. The new method is based on the dyadic wavelet transform and utilizes two FIR-filters for its implementation. The construction of an analysis wavelet with certain criteria; the quadratic spline wavelet, is explained in detail. The proposed algorithm has been implemented on a wide range of musical signals and the results are compared to those obtained by the autocorrelation function.

From the implementation and the simulation results, we conclude that the proposed algorithm is quite simple since only two FIR-filters are required for the analysis. These are the low-pass filter corresponding to the scale function and the high-pass filter corresponding to the analysis wavelet. The proposed algorithm has proven to be accurate, efficient and robust to noise.

The performance of various wavelets has been tested and the quadratic spline wavelet is proven to be appropriate for the analysis of a wide variety of musical signals. In addition, the frame size can be chosen arbitrarily as long as the minimum number of periods is preserved for estimating the pitch period. Therefore we conclude that the choice of either the analysis wavelet or the frame size is not as crucial as it is in the autocorrelation function. As previously mentioned the computational complexity of the proposed algorithm is $O(NJ)$ where N is the size of the input signal and J is the total number of scales. The main advantage of the proposed technique is that it is faster than the auto-correlation function and takes into account the non-stationary nature of musical signals. Finally the proposed algorithm is general in the sense that it can be applied not only to guitar signals but also to a wide class of musical signals.

Having estimated the pitch period, the next step in the analysis is to study the time-varying spectrum of the guitar signal. This step provides the partials'

decay rates which consequently determine the magnitude response of the loop filter. Chapter 4 has presented a thorough study of this step. We start by describing one of the most commonly used techniques in the literature; that is the STFT-based analysis. Implementation aspects and simulation results using this technique are presented in detail. The main limitations of this technique are briefly described.

The novelty of chapter 4 stems from introducing a new technique for partials tracking in musical signals. The proposed technique is based on the theory of Kalman filter in non-linear systems. Different models have been tested, the implementation aspects of the algorithm are thoroughly presented and the simulation results are analysed in depth.

From this chapter we conclude that designing the correct model that fits the true signal is the core of the EKF technique. Once the model has been designed accurately, the computation of the state transition matrix and the measurement matrix is straightforward.

The results of the proposed technique show that it is accurate and robust to noise. The computational complexity of this technique is $O(M^2)$ where M is the total number of state variables to be estimated. This implies that the EKF-based technique is significantly faster than the STFT-based technique which performs a series of FFT each of complexity $O(N \log N)$ for an N-point input signal. Unlike the STFT-based technique which implements a peak detection and a peak continuation algorithm to provide the trajectories of each partial, the EKF-based technique provides the trajectories of each partial in one step. Further, the phase envelopes are obtained by linear interpolation instead of cubic interpolation as is the case in the STFT-based technique.

Another major advantage of the proposed technique over the STFT-based technique is the time-resolution. The fact that the maximum value of the time increment parameter D is limited by the sampling theorem indicates that this

technique takes into account the non-stationary nature of the musical signals. Hence the results of this technique can be considered as more reliable than the ones obtained by the STFT technique. It is worth mentioning here that due to the time-resolution problem of the STFT-based analysis, the whole process has to be carried out once more for extracting the lowest body resonances. In contrast, the lowest body resonances have already been included in the EKF model.

The proposed EKF-based algorithm has been tested mainly for guitar signals. However we believe that the algorithm can be applied to any musical signal that can be represented by the final model; for instance any plucked string signal. Nevertheless, the application of the proposed technique to other types of musical signals is ripe for further investigations.

On the other hand some parameters of the EKF-based algorithm have to be carefully chosen such as the measurement noise variance R and the state covariance matrix Q. A prior knowledge of the signal to be analysed can be exploited. In our study the values of R and Q have been experimentally chosen.

At this point it is worth noting that all proposed techniques have been implemented using MATLAB. They have the potential of real-time implementation and writing these algorithms in a more generic language like C is currently taking place. Having estimated the required values for calibrating the guitar model, the estimation process is explained in detail in chapter 5. we believe that this chapter is of great importance as it assures the accuracy of the proposed EKF-based technique. In chapter 5 we have also presented an analysis of the re-synthesized signal as well as the EKF-residual signal.

6.2 Future Work

We believe that the topics introduced in this dissertation are just the beginning of a new era for digital sound synthesis techniques. There are still many things to

be done in both the modelling process and the analysis process. In the modelling process, modelling the non-linearities present in real strings needs more careful study and new techniques for incorporating them are required. Modelling the guitar body as a digital filter as well as modelling the plucking point are still opened for further investigations. It is possible that the wavelet technique we have introduced may form a starting point for this difficult problem. We also believe that new techniques are required for the calibration process. This includes the design of the loop filter, the choice of the optimal weight function and the estimation of the excitation signals; more experimental and theoretical work is needed to clarify this area.

The EKF technique shows great promise in the modelling of instruments. Nonetheless a method for computing the state covariance matrix Q in musical signals is required. Also the application of the EKF technique to a wider range of musical signals needs to be studied, but we have shown its potential. The main obvious continuation of this study is to adapt the pitch detection algorithm as well as the partials tracking method for the SMS modelling system. We believe that these methods will lead to significant improvement in this technique, and offer the promise of real-time analysis.

6.3 Conclusions

To conclude, in this study we have presented novel methods for the analysis of musical signals. A new pitch detection algorithm is developed which has the potential of real-time implementation and is proven to be efficient and accurate. Another technique is developed for tracking the partials in plucked string signals, this new technique being simple, accurate and provides a substantial reduction of the computational complexity compared to the STFT-based technique.

Appendix A

Paper from DAFx99

This appendix is a copy of the paper *A Wavelet-based Pitch Detector for Musical Signals* by J. ffitch and W. Shabana, which was published in the Proceedings of DAFx99 [10].

A WAVELET-BASED PITCH DETECTOR FOR MUSICAL SIGNALS

John Fitch
Department of Mathematical Sciences,
University of Bath,
Bath BA2 7AY, UK
e-mail: jpff@maths.bath.ac.uk

Wafaa Shabana
Department of Mathematical Sciences,
University of Bath,
Bath BA2 7AY, UK
e-mail: wrs@maths.bath.ac.uk

ABSTRACT

Physical modelling of musical instruments is one possible approach to digital sound synthesis techniques. By the term physical modelling, we refer to the simulation of sound production mechanism of a musical instrument, which is modelled with reference to the physics using wave-guides. One of the fundamental parameters of such a physical model is the pitch, and so pitch period estimation is one of the first tasks of any analysis of such a model.

In this paper, an algorithm based on the Dyadic Wavelet Transform has been investigated for pitch detection of musical signals. The wavelet transform is simply the convolution of a signal f(t) with a dialated and translated version of a single function called the mother wavelet that has to satisfy certain requirements. There are a wide variety of possible wavelets, but not all are appropriate for pitch detection. The performance of both linear phase wavelets (Haar, Morlet, and the spline wavelet) and minimum phase wavelets (Daubechies' wavelets) have been investigated. The algorithm proposed here has proved to be simple, accurate, and robust to noise; it also has the potential of acceptable speed. A comparative study between this algorithm and the well-known autocorrelation function is also given. Finally, illustrative examples of different real guitar tones and other sound signals are given using the proposed algorithm.

KEYWORDS

Physical modeling – wavelet transform – pitch – autocorrelation function.

1. INTRODUCTION

During the last two decades, physical modelling of real musical instruments has gained popularity as a tool for sound synthesis and computer music. The term physical modelling refers to the simulation of sound production mechanism and the behaviour of a real musical instrument [1] [2] [3].

In physical modelling of a guitar (as a plucked string instrument), the ideal vibrating string is considered as the main source of vibration. It satisfies the one-dimensional wave equation, which can be modelled very accurately using digital wave-guide techniques [4]. Starting with a recorded real guitar tone, estimating the model parameters is one of the main tasks of the analysis process. Hence, pitch period estimation is essential for extracting the other parameters. Unlike speech signals, musical signals have a broader range of frequencies, so there are some difficulties in estimating their pitch period [5]. The autocorrelation function is one of the well-known time-domain pitch detectors. Despite its simplicity, the autocorrelation function has some disadvantages.

An algorithm based on the dyadic wavelet transform has been investigated for pitch estimation of musical signals. The basic idea of this algorithm is that, for an appropriately chosen wavelet, the dyadic wavelet transform exhibits local maxima at the points of sharp variation of the signal [6].

In this paper, the application of the proposed algorithm to a wide range of stringed musical signals as well as some other musical signals has been investigated. Further, a comparative study between this algorithm and the autocorrelation function is presented. This paper is organized as follows: section 2 is devoted to the pitch detection problem and the autocorrelation algorithm. In section 3, principles of the dyadic wavelet transform and its properties is presented. In section 4, implementation of the proposed algorithm and the autocorrelation algorithm to a wide range of musical signals as well as singing voices is studied. Discussions and results are presented in section 5. Finally, section 6 is devoted to the conclusion.

2. PITCH DETECTION OF MUSICAL SIGNALS

Pitch period is a fundamental parameter in the analysis process of any physical model. A pitch detector is basically an algorithm that determines the fundamental pitch period of an input musical signal. Pitch detection algorithms can be divided into two groups: time-domain pitch detectors and frequency-domain pitch detectors. Pitch detection of musical signals is not a trivial task due to some difficulties such as the attack transients, low frequencies, and high frequencies.

The autocorrelation function is a time-domain pitch detector. It is a measure of similarity between a signal and translated (shifted) version of itself. The basic idea of this function is that periodicity of the input signal implies periodicity of the autocorrelation function and vice versa.

For non-stationary signals, short-time autocorrelation function for signal $f(n)$ is defined as [7]:

$$ph(m) = \frac{1}{N} \sum_{n=0}^{N-m-1} [f(n+l)w(n+l)][f(n+m+l)w(n+m+l)], \quad (1)$$

$$0 \le m \le M_0 - 1,$$

where $w(n)$ is an appropriate window function, N is the frame size, l is the index of the starting frame, m is the autocorrelation

parameter or time lag and M_0 is the total number of points to be computed in the autocorrelation function.
The autocorrelation function has its highest peak at m=0 which equals to the average power of the input signal. For each l, one searches for the local maxima in a meaningful range of m. The distance between two consecutive maxima is the pitch period of the input signal $f(n)$. Different window functions such as rectangular, Hanning, Hamming, and Blackman windows have been used in the analysis. The choice of an analysis window and the frame size are among the main disadvantages of the autocorrelation function.

3. DYADIC WAVELET TRANSFORM

Wavelet transform is based on the idea of filtering a signal f(t) with a dialated and translated versions of a prototype function $\Psi(t)$. This function is called the mother wavelet and it has to satisfy certain requirements [8]. The *Continuous Wavelet Transform* (CWT) for f(t), is defined as [9];

$$CWT(f,a,b) = \int_{-\infty}^{\infty} f(t)\Psi_{a,b}(t)dt\,, \qquad (2)$$

where $\Psi_{a,b}(t) = \Psi(\frac{t-b}{a})$, $a \in R-\{0\}$ is the scale parameter and $b \in R$ is the translation parameter. In addition to its simple interpretation, the CWT satisfies some other useful properties such as linearity and conservation of energy [8] [9]. For practical implementations, CWT is computationally very complex.
Dyadic Wavelet Transform (DWT), is the special case of CWT when the scale parameter is discretized along the dyadic grid (2^j), j=1, 2... and $b \in Z$ [10], i.e.,

$$\text{DWT(f,j)} = W_j f = f(t) * \Psi_{2^j}(t) \qquad (3)$$

where * denotes convolution and $\Psi_{2^j}(t) = \frac{1}{2^j}\Psi(\frac{t}{2^j})$. For an appropriately chosen wavelet, the wavelet transform modulus maxima denote the points of sharp variations of the signal [6] [10] [12]. This property of DWT has been proven very useful for detecting pitch periods of speech signals [11]. An appropriately chosen wavelet is a wavelet that is the first derivative of a smooth function [6]. Zero-crossings of musical signals can be considered as points of sharp variation of the signal and hence the dyadic wavelet transform exhibits local maxima at these points across several consecutive scales. The pitch period is evaluated by measuring the time distance between two such consecutive maxima.

4. IMPLEMENTATION

Theoretically, the *Dyadic wavelet transform* has to be evaluated for all scales (2^j), for j varying from $-\infty$ to $+\infty$. For practical implementation, one is limited to a finite larger scale and a nonzero finer scale, since the input signal is generally measured with a finite resolution. The finer scale is equal to 1 (for normalization purposes) and the larger scale is equal to 2^J. The wavelet used in this analysis is the quadratic spline wavelet, which is the first derivative of the cubic spline $\theta(t)$, i.e.,

$$\hat{\Psi}(\omega) = i\omega\hat{\theta}(\omega), \qquad (4)$$

with

$$\hat{\theta}(\omega) = \exp(-i\omega)\left(\frac{\sin(\omega/4)}{(\omega/4)}\right)^4 \qquad (5)$$

This wavelet is an anti-symmetric, regular and of compact support. The corresponding scale function $\Phi(t)$ is the quadratic spline with Fourier transform given by

$$\hat{\Phi}(\omega) = \exp(-i\frac{3\omega}{2})\left(\frac{\sin(\omega/2)}{(\omega/2)}\right)^3 \qquad (6)$$

Figure 1. show both $\Psi(t)$ and $\Phi(t)$ respectively.

Two FIR filters, namely, a low-pass filter $\{h(n)\}$ and a high-pass filter $\{g(n)\}$ characterize the discrete dyadic wavelet transform, and the number of levels J. Starting with $S_0 f = f$, $h_0 = h$ and $g_0 = g$, the recursive algorithm is defined as

$$\left.\begin{aligned} W_{j+1} f &= g_j * S_j f \\ S_{j+1} f &= h_j * S_j f \end{aligned}\right\}, \qquad (7)$$

j=0,1...J-1.

where h_j and g_j denotes the filters obtained from h and g by inserting $2^j - 1$ zeros between each two consecutive coefficients of the two filters respectively. Hence the DWT can be implemented as a FIR non-subsampled octave-band filter bank.

(a) quadratic spline wavelet (b) quadratic spline function

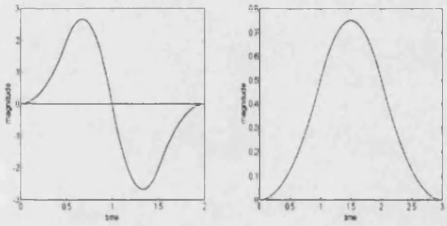

Figure (1)

5. RESULTS AND DISCUSSONS

The proposed algorithm has been implemented on a wide range of musical signals such as a saxophone signal (wind instrument), a tanpura signal (an Indian drone instrument), a singing voice signal, and a conga rim-hit signal (drum family) but with emphasis on plucked string signals (a classical guitar tone, bass, pizzacato cello ...etc).
The sampling rate for all test signals is 44.1 kHz and different window size of 22.7 and 34 ms has been used. Experiments have shown that it is adequate to evaluate the dyadic wavelet transform across three consecutive scales only 2^4, 2^5, and 2^6.
In the analysis of plucked string signals, the results show that it is sufficient to estimate the pitch period from the steadily decaying part of the signal several hundred milliseconds after the attack [13]. This is due to the fact that pitch period of plucked-string signal decreases as the signal attenuates. The test signal is a D-tone guitar and the estimated pitch period is 147 Hz. In this case, the relative error is (0.001). Results for guitar tone is shown in figure (2).

The estimated pitch of a D#-sax signal is 154.7368 Hz with relative error 0.005. Figure (3) shows results of the sax signal.
The tanpura signal is a very harmonically rich signal. Our results show that the proposed algorithm has the ability to detect not only the fundamental frequency but also the frequency with the most energy present in the signal. In the case of the autocorrelation function, the effect of the window function is to taper the function smoothly to 0. Hence, a longer frame size has to be used in order to detect the fundamental of the tanpura signal not the strongest harmonic. The estimated pitch period of this signal is 157.5 Hz. Results of this signal are shown in figure (4).
Figure (5) shows results of the analysis of a male singing voice. The estimated pitch is 110.8040 Hz.
Moreover, in the analysis of conga-rim signal, the algorithm classified this signal as unpitched one since it failed to find local maxima that satisfy the previous criteria.
For all test signals, the results can be further improved by using several methods of curve fitting for best estimate of local maxima.
The computational complexity of the proposed algorithm is O(NJ), for an input signal of length N evaluated across J scale. The constant depends on the number of the nonzero coefficients present in the filters h and g. The algorithm is faster than the autocorrelation function since the length of the analysis wavelet is less than M_0.
Different wavelets like Haar wavelet [14], a minimum-phase wavelet [14], and Morlet wavelet [15], have been used in the analysis to compare their performance. The spline wavelet has a superior performance. Results also show that the Haar wavelet has the potential of real-time implementation due to its simplicity and its accurate results.
Despite its simplicity, the autocorrelation function is computationally expensive when the appropriate frame size is used. Its main drawback is the choice of a window function and assuming stationarity of the signal within the frame, hence using a fixed frame size during the analysis process. More about the analysis process of all test signals is found in [16].

6. CONCLUSIONS

An algorithm based on the Dyadic Wavelet Transform is investigated for pitch detection of musical signals. The results show that the algorithm can be applied to a wide range of musical signals such as guitar, sax, cello, bass, tanpura as well as some singing voices. The algorithm is simple since only two FIR-filters are required for the analysis. It is accurate, efficient and robust to noise. The main advantage of the proposed algorithm is that it is fast compared to the autocorrelation function. Besides, the algorithm takes into account the non-stationarity of the input signal. Unlike the autocorrelation function, the frame size in not a crucial parameter since different frame sizes have been used successfully. On comparing the performance of different wavelets, the quadratic spline wavelet has a superior performance. Nevertheless, the algorithm has the potential of real-time implementation using the Haar wavelet due to its simplicity with minimal loss of accuracy. Finally, the algorithm can classify unpitched signals.

REFERENCES

[1] Valimaki, V. and Takala, T., "Virtual Musical Instruments-natural sounds using physical models", Organized Sound, 1(2): 75-86, (1996).

[2] Valimaki, V., Huopaniemi, J., Karjalainen, M. and Janosy, Z., "Physical modeling of plucked string instruments with appl-ication to real-time sound synthesis", Journal of the Audio Engineering Society, 44(5): 331-353, (1996).

[3] Karjalainen, M., Valimaki, V. and Tolonen, T., "Plucked string models-from Karplus-strong algorithm to digital waveguides and beyond", Computer Music Journal, 22(3): 17-32, (1998).

[4] Smith, J.O., " Physical modeling using digital waveguides", Computer Music Journal, 16(4):74 -91, (1992).

[5] Roads, C., " The computer music tutorial " , The MIT Press, Cambridge, Massachusetts, USA, 504-520, (1995).

[6] Mallat S. and Zhong S., " Characterization of signals from multiscale edges", IEEE Trans. on Pattern Analysis and Machine intelligence, 14(7): 710-732, (1992).

[7] Rabinar L R. , " On the use of autocorrelation analysis for pitch detection", IEEE Trans. on Acoustics, Speech and Signal Processing, 25(1):24- 33, (1977).

[8] Kronland Martinet, R. , Morlet, J. and Grossman, A., "Analysis of sound patterns through wavelet transforms", International Journal of Pattern Recognition and Artificial Intelligence, 1(2): 273-302, (1987).

[9] Sadowsky, J., "The continuous wavelet transform: a tool for signal investigation and understanding", Johns Hopkins APL Technical Digest, 15(4): 306-318, (1994).

[10] Mallat, S., " Zero- crossing of a wavelet transform " , IEEE Trans. on Information Theory, 37(4): 1019-1033, (1991).

[11] Kadambe, S., and Boudreaux-Bartels, G., "Application of the wavelet transform for pitch detection of speech signals", IEEE Trans. on Information Theory, 38(2): 917-924, (1992).

[12]Berman, Z. and Baras, J. S., " Properties of the multiscale maxima and zero-crossings representations", IEEE Trans.on Signal Processing, 41(12): 3216-3231, (1993).

[13]Tolonen, T., " Model-based analysis and resynthesis of Acoustic Guitar Tones", Helsinki University of Technology Department of Electerical and Communications Engineering, Technical report 46, (1998).

[14]Daubechies, I., "Orthonormal bases of compactly supported wavelets", Commun.Pure and Applied Math, 41: 909-996, (1988).

[15]Kadambe, S., and Boudreaux-Bartels, G., "A comparison of wavelet functions for pitch detection of speech signals", Proc. of ICASSP, May 449-452, (1991).

[16]Shabana, W., " A wavelet-based pitch detector for musical signals " Technical report, Department of Mathematical Sciences, University of Bath, Oct., (1999).

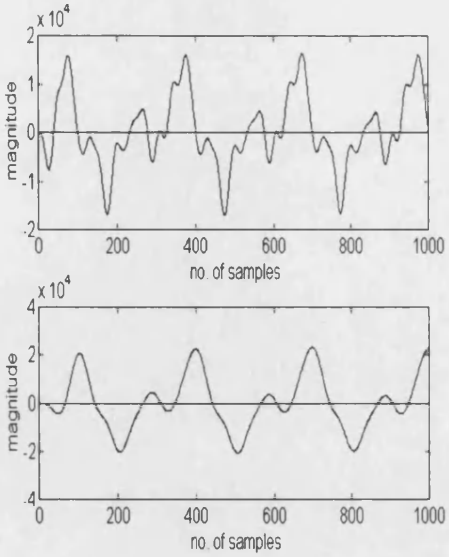

Figure 2. scale 2^5 and scale 2^6 of a of the guitar tone using quadratic spline wavelet

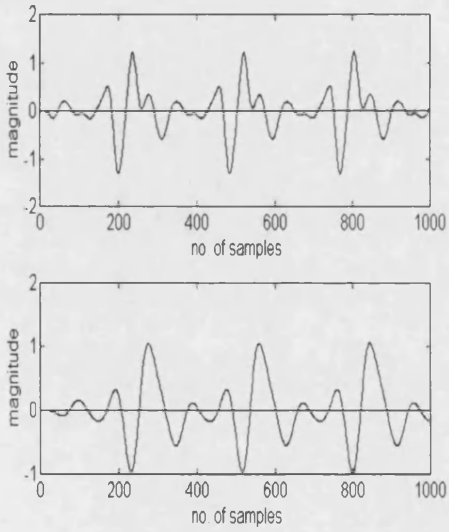

Figure 3. top: scale 2^5 and bottom: scale 2^6 of the sax signal using quadratic spline wavelet

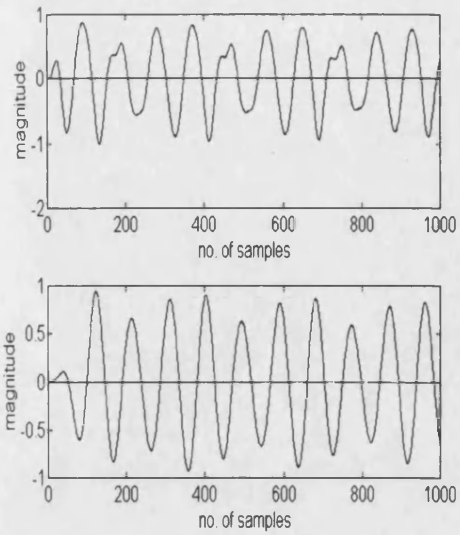

Figure 4. Scale 2^5 (top) and scale 2^6 (bottom) of a tanpura signal using quadratic spline wavelet

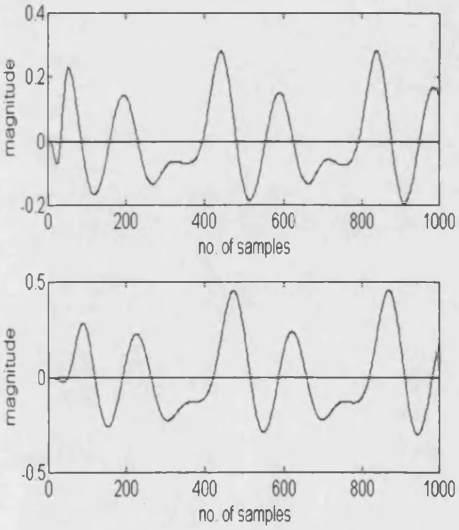

Figure 5. top: scale 2^5 and bottom: scale 2^6 of a singing voice signal using quadratic spline wavelet

Appendix B

Parabolic Interpolation

The estimated values of the local maxima for the autocorrelation function and the wavelet-based pitch detector can be further improved by using parabolic interpolation. This is achieved by fitting a parabola to the maximum point, say $A(x_2, y_2)$, and its two adjacent points $B(x_1, y_1)$ and $C(x_3, y_3)$.
The equation of the parabola is given by

$$y = a\,x^2 + b\,x + c \tag{B.1}$$

where a, b and c are the coefficients of the parabola that need to be estimated. On substituting from A, B, and C into equation (B.1) yields

$$y_1 = a\,x_1^2 + b\,x_1 + c \tag{B.2}$$

$$y_2 = a\,x_2^2 + b\,x_2 + c \tag{B.3}$$

$$y_3 = a\,x_3^2 + b\,x_3 + c \tag{B.4}$$

Subtract equation (B.2) from equations (B.3) and (B.4)

$$y_2 - y_1 = a\,(x_2^2 - x_1^2) + b\,(x_2 - x_1) \tag{B.5}$$

$$y_3 - y_1 = a\,(x_3^2 - x_1^2) + b\,(x_3 - x_1) \tag{B.6}$$

Equations (B.5) and (B.6) can be rewritten as

$$I = a(x_2 + x_1) + b \tag{B.7}$$

$$II = a(x_3 + x_1) + b \tag{B.8}$$

where I and II are defined as

$$I = \frac{(y_2 - y_1)}{(x_2 - x_1)} \tag{B.9}$$

$$II = \frac{(y_3 - y_1)}{(x_3 - x_1)} \tag{B.10}$$

To determine the coefficient a, subtract equations (B.7) and (B.8)

$$a = \frac{(I - II)}{(x_2 - x_3)} \tag{B.11}$$

Using the value of a, the coefficient b can be determined from equation (B.7) as

$$b = I - a(x_2 + x_1) \tag{B.12}$$

By knowing the values of both a and b, the coefficient c is determined from any of equations (B.2), (B.3), or (B.4). The maximum value of the parabola then evaluates to

$$x_{max} = \frac{-b}{2a} \tag{B.13}$$

$$y_{max} = c - \frac{b^2}{4a} \tag{B.14}$$

Appendix C

Analysis of a D Guitar Signal

In this appendix we present the analysis of another guitar test signal which is shown in figure (C.1). The estimated pitch period of this signal using the quadratic spline wavelet is 147.027 Hz. The results are shown in figure (C.2).

Figure (C.3) shows the Fourier transform of this signal. From this figure it is clear that the second body resonance is too small (nearly missing). Therefore in the EKF-based model we have included only the first body resonance.

Some of the amplitude trajectories of this signal using the STFT-based technique are shown in figure (C.4). Figure (C.5) shows the corresponding results of the proposed EKF-based technique.

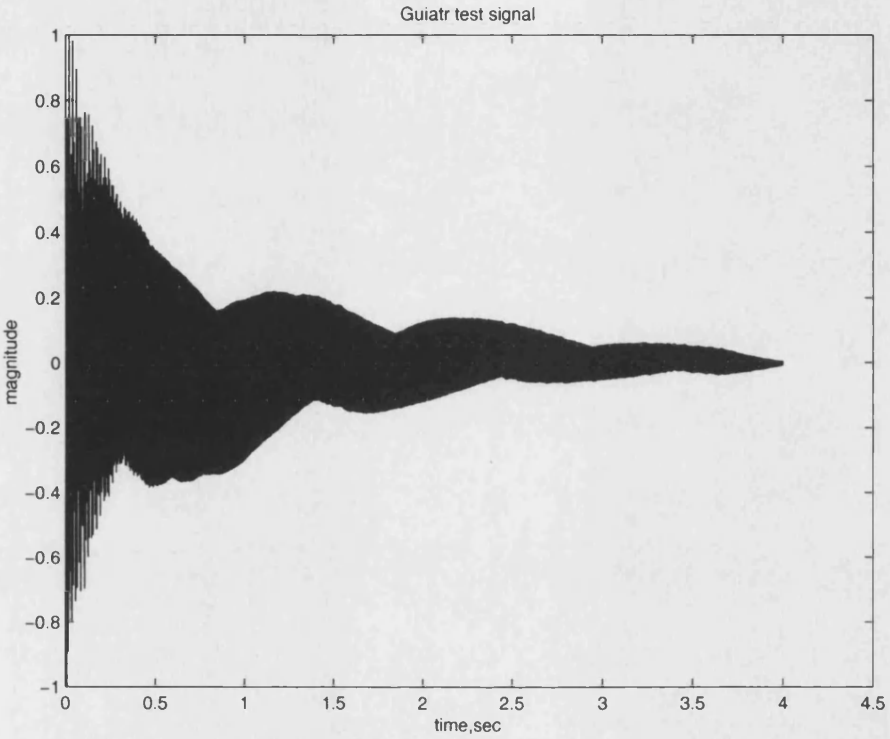

Figure C.1: Guitar test signal

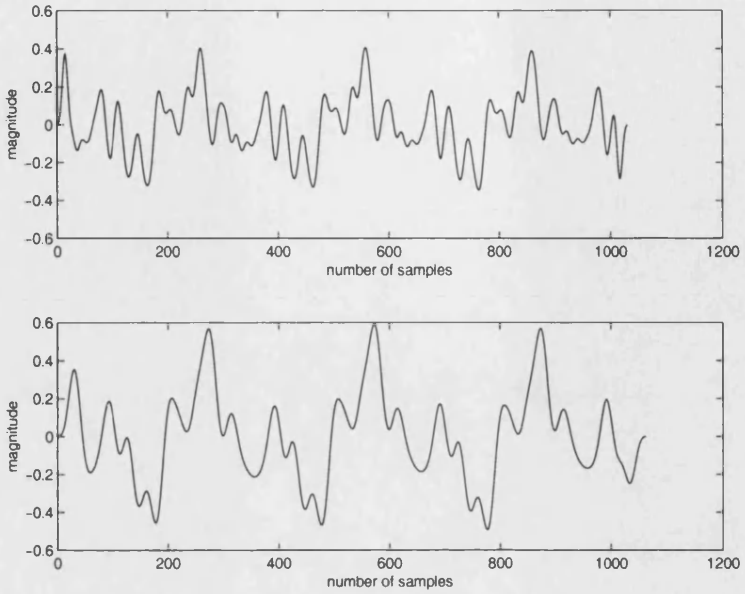

Figure C.2: Scales 2^4 and 2^5 of the guitar signal

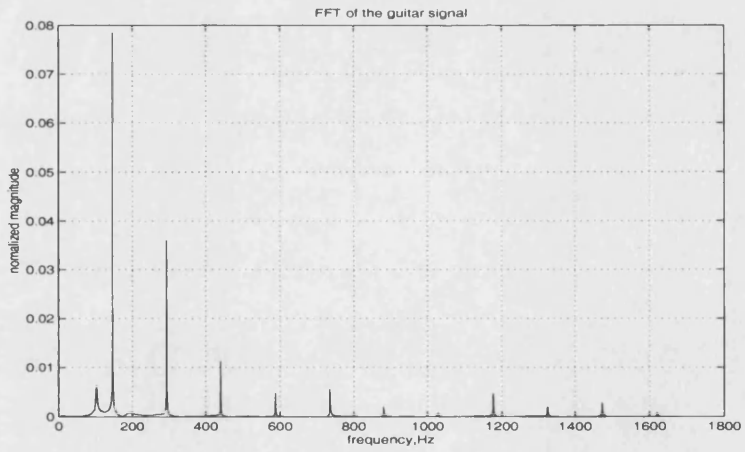

Figure C.3: FFT of the D-guitar test signal

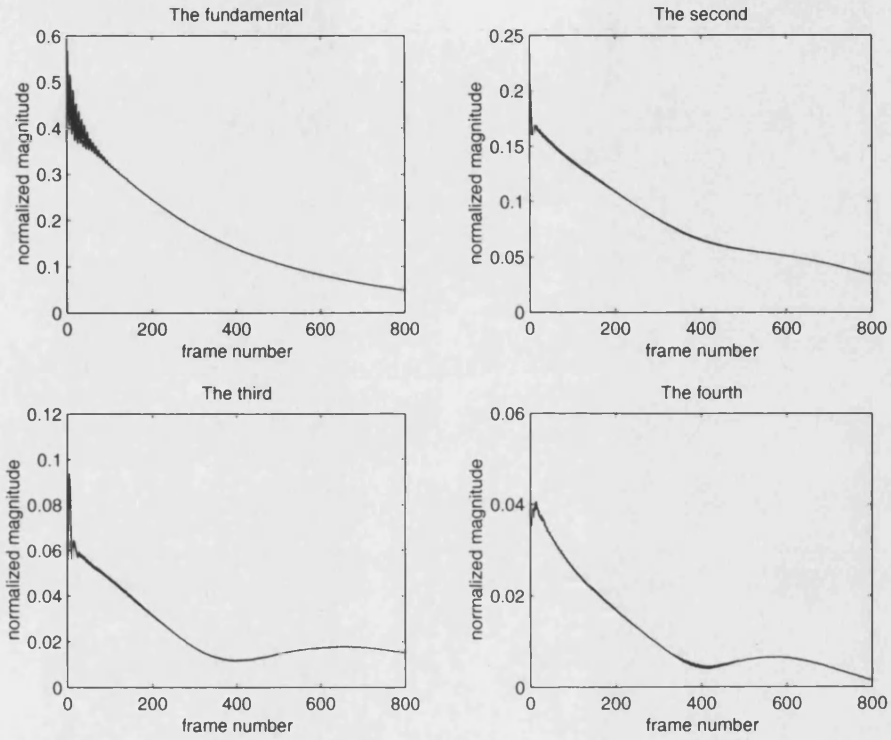

Figure C.4: Amplitude trajectories of the D-guitar signal using the STFT technique

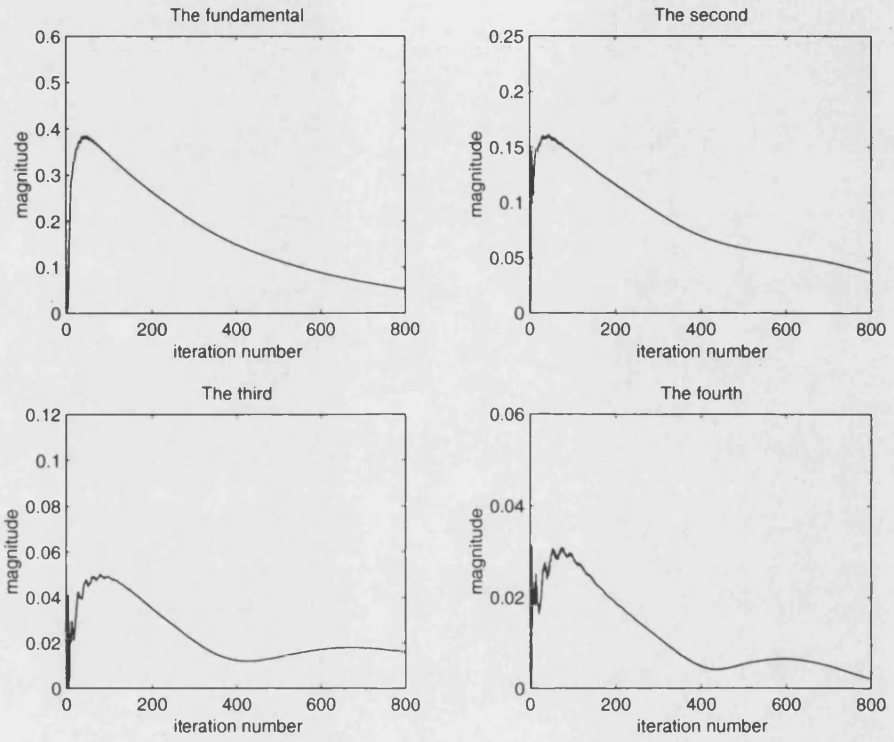

Figure C.5: Amplitude trajectories of the D-guitar signal using the EKF technique

Appendix D

Analysis of a B Guitar Signal

In this appendix we present the analysis of another guitar test signal which is shown in figure (D.1). The estimated pitch period of this signal using the quadratic spline wavelet is 248.1836Hz. The results are shown in figure (D.2).

Figure (D.3) shows the Fourier transform of this signal. From this figure it is clear that partials above 3kHz are too small to be included in the EKF-based model. In addition, we have included only the first body resonance since the second is nearly missing.

The amplitude trajectories of the four lowest partials of the guitar signal using the STFT-based analysis are shown in figure (D.4).

Figure (D.5 shows the trajectories of these partials using the EKF-based technique.

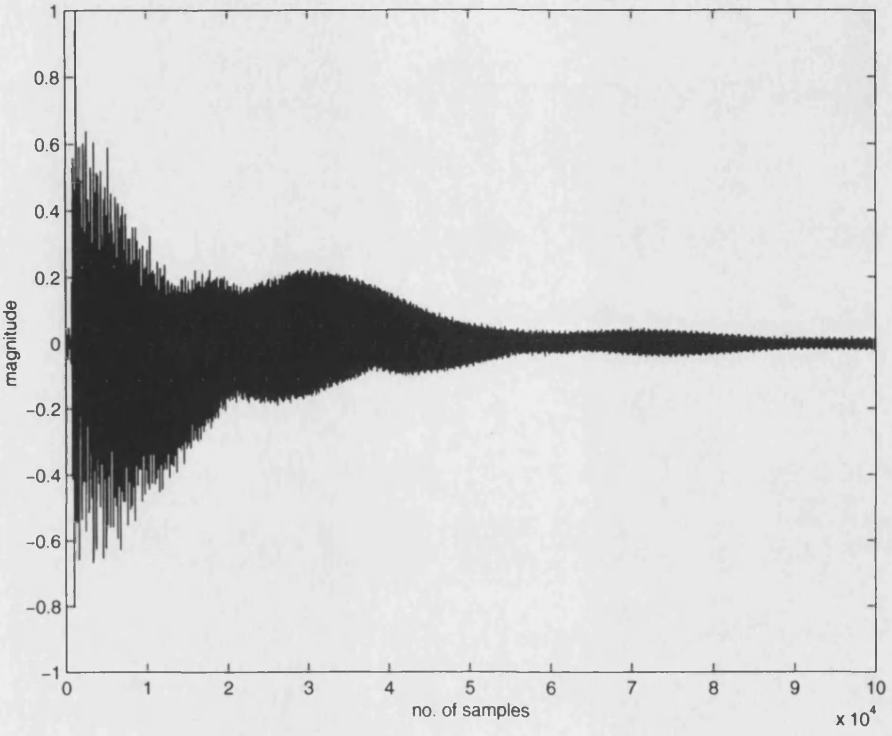

Figure D.1: Guitar test signal B1

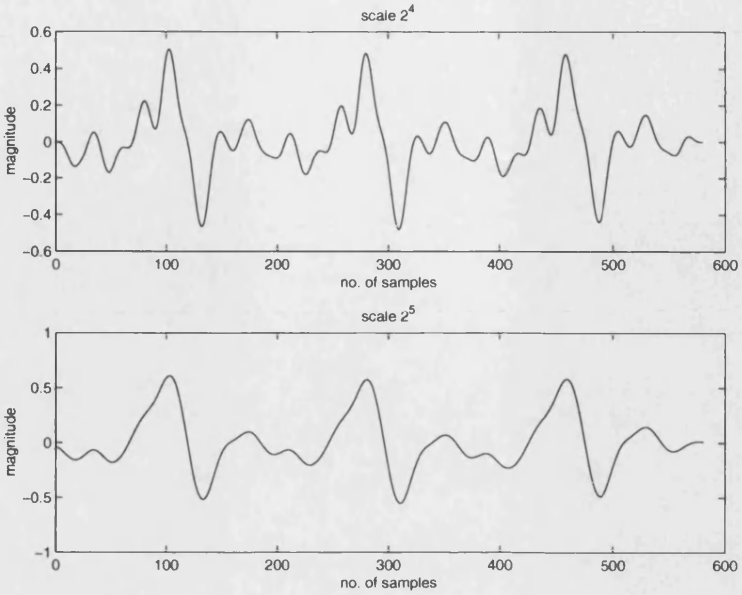

Figure D.2: Scales 2^4 and 2^5 of the guitar signal

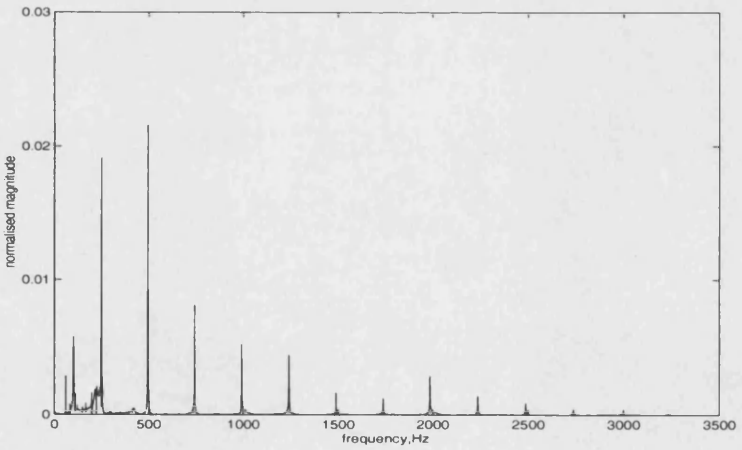

Figure D.3: FFT of the B1-guitar test signal

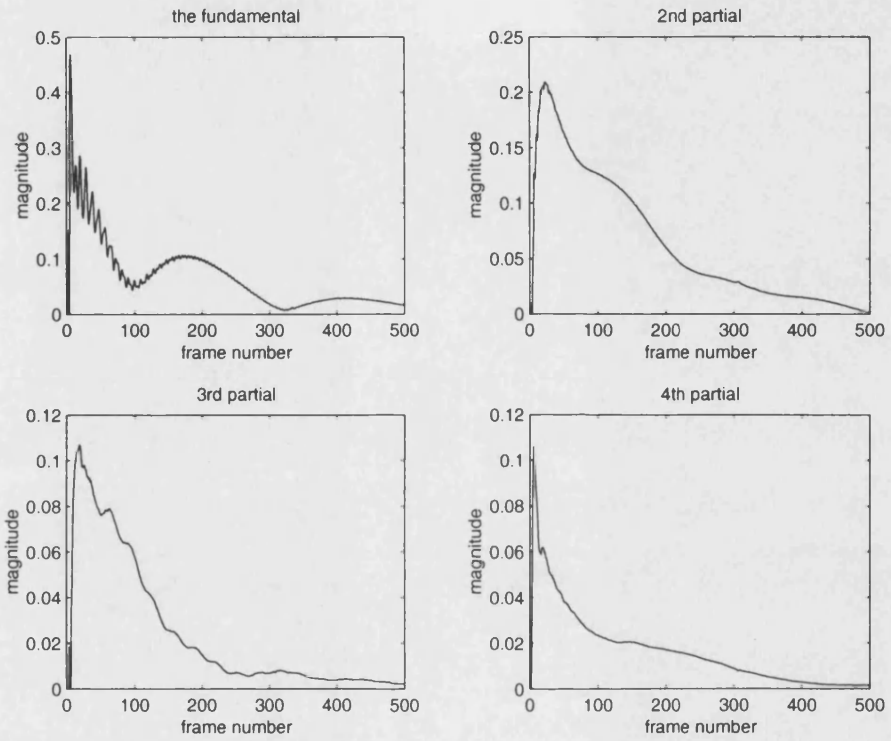

Figure D.4: Amplitude trajectories of the B1-guitar signal using the STFT technique

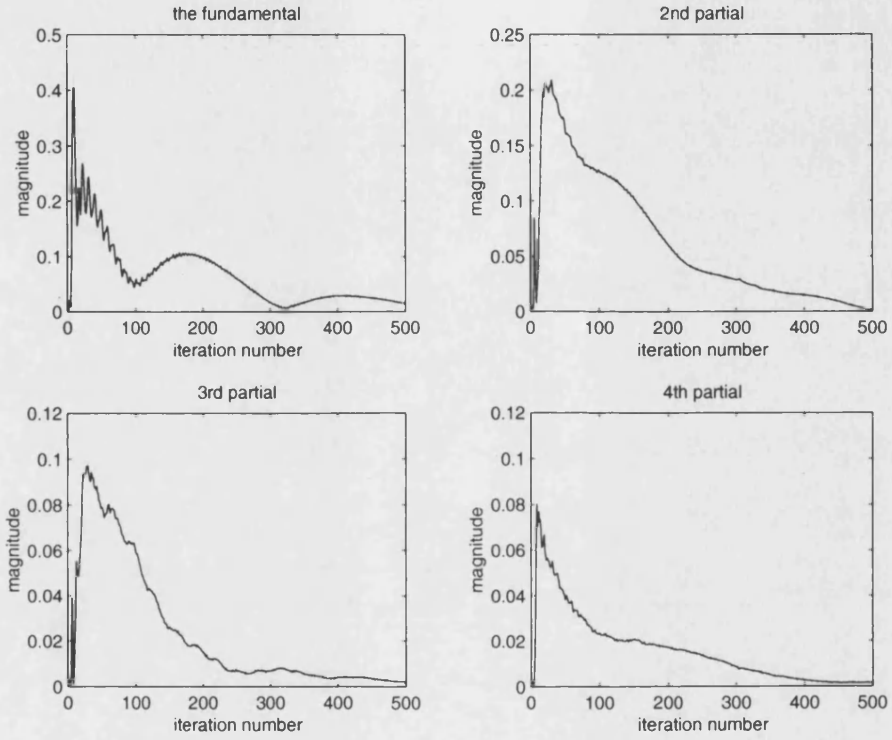

Figure D.5: Amplitude trajectories of the B1-guitar signal using the EKF-technique

Bibliography

[1] J. B. Allen. Short term spectral analysis, synthesis, and modification by discrete Fourier transform. *IEEE transactions on Acoustics, Speech, and Signal Processing*, 25(3):235–238, 1977.

[2] J. B. Allen and L. R. Rabiner. A unified approach to short-time Fourier analysis and synthesis. *Proc. IEEE*, 65:1558–1564, 1977.

[3] Z. Berman and J. S. Baras. Properties of the multiscale maxima and zero-crossings representations. *IEEE Trans. on Signal Processing*, 41(12):3216–3231, 1993.

[4] S. M. Boaiz. *Digital and Kalman filtering: an introduction to discrete-time filtering and optimum linear estimation.* Edward Arnold, 1979.

[5] R. G. Brown and P. Y. C. Hwang. *Introduction to random signals and applied Kalman filtering.* John Wiley and Sons, Inc., 2nd edition, 1992.

[6] P. R. Cook. Non-linear periodic prediction for on-line identification of oscillator characteristics in woodwind instruments. In *Proceedings of the ICMC, Montreal*, pages 157–160, 1991.

[7] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Commun. Pure and Applied Math*, 41:909–996, 1988.

[8] I. Daubechies. *Ten lectures on wavelets.* SIAM, Philadelphia, PA, 1992.

[9] N. H. Feltcher and T. D. Rossing. *The physics of musical instruments.* Springer-Verlag, New York, USA, 1991.

[10] J. ffitch and W. Shabana. A wavelet-based pitch detector for musical signals. In Jan Tro and Mikael Larsson, editors, *Proceedings of DAFx99*, pages 101–104. Department of Telecommunications, Acoustics Group, Norwegian University of Science and Technology, December 1999.

[11] J.-L. Florens and C. Cadoz. The physical model: modelling and simulating the instrument universe. In G. D. Poli, A. Piccialli, and C. Roads, editors, *Representations of Musical Signals.* The MIT Press, Cambridge, Massachusetts, USA, 1991.

[12] A. A. Girgis and W. L. Peterson. Adaptive estimation of power system frequency deviation and its rate of change for calculating sudden power system overloads. *IEEE Transactions on Power Delivery*, 5(2):585–594, 1990.

[13] A. Haar. Zur theorie der orthogonalen funktionen systeme. *Math. Annal.*, 69:331–371, 1910.

[14] F. J. Harris. On the use of the windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66:51–83, 1978.

[15] J. Hiipakka, V. Valimaki, and M. Karjalainen. Implementation and control of a real-time guitar synthesizer. Master's thesis, Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Espoo, Finland, 1999.

[16] L. Hiller and P. Ruiz. Synthesizing musical sounds by solving the wave equation for vibrating objects: part(I). *Journal of the Audio Engineering Society*, 19(6):462–470, 1971.

[17] L. Hiller and P. Ruiz. Synthesizing musical sounds by solving the wave equation for vibrating objects: part(II). *Journal of the Audio Engineering Society*, 19(7):542–550, 1971.

[18] D. A. Jaffe and J. O. Smith. Extensions of the Karplus-Strong plucked-string algorithm. *Computer Music Journal*, 7(2):76–87, 1983.

[19] J. M. Jot. An analysis/synthesis approach to real-time artificial reverbration. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol 2*, pages 221–224, 1992.

[20] S. Kadambe and G. Bourdreaux-Bartels. A comparison of wavelet functions for pitch detection of speech signals. In *Proc. of ICASSP*, pages 449–452, May 1991.

[21] S. Kadambe and G. Bourdreaux-Bartels. Applications of the wavelet transform for pitch detection of speech signals. *IEEE Trans. on Information Theory*, 38(2):917–924, 1992.

[22] M. Karjalainen and U. K. Laine. A model for real-time sound synthesis of guitar on a floating-point signal processor. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3653–3656, Toronto, Canada, 1991.

[23] M. Karjalainen and J. O. Smith. Body modeling techniques for string instrument synthesis. In *Proceedings of the International Computer Music Conference, Hong Kong*, pages 232–239, 1996.

[24] M. Karjalainen, V. Valimaki, J. Huopaniemi, and Z. Janosy. Physical modeling of plucked string instruments with application to real-time sound synthesis. *Journal of the Audio Engineering Society*, 44(5):331–353, 1996.

[25] K. Karplus and A. Strong. Digital synthesis of plucked-string and drum timbers. *Computer Music Journal*, 7(2):43–55, 1983.

[26] R. Kronland Martinet, J. Morlet, and A. Grossman. Analysis of sound patterns through wavelet transforms. *International Journal of Pattern Recognition and Artificial Intelligence*, 1(2):273–302, 1987.

[27] S. Mallat. Zero-crossings of a wavelet transform. *IEEE Trans. on Information Theory*, 37(4):1019–1033, 1991.

[28] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 14(7):710–732, 1992.

[29] P. Masri. *Computer modelling of sound for transformation and synthesis of musical signals.* PhD thesis, Digital Music Research Group, University of Bristol, 1996.

[30] P. S. Maybeck. Stochastic models, estimation and control. *Mathematics in Science and Engineering*, 141, 1979.

[31] R. J. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(6):744–754, 1986.

[32] F. R. Moore. *Elements of computer music.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1990.

[33] P. M. Morse. *Vibration and Sound.* The American Institute of Physics, 1986.

[34] P.M. Morse and U. Ingrad. *Theoretical Acoustics.* McGraw-Hill, New York, 1968.

[35] A. H. Nuttal. Some windows with very good side-lobe behaviour. *IEEE transactions on Acoustics, Speech, and Signal Processing*, 29(1):84–91, 1981.

[36] S. J. Orfanidis. *Introduction to Signal Processing.* Prentice Hall, New Jersey, USA, 1996.

[37] G. D. Poli, S. Canazza, A. Bari, and G. A. Mian. Improving the extended Kalman filter method for the restoration of electro-acoustic music. In *Proceedings of the ICMC 2000, Berlin, Germany*, pages 233–236, 2000.

[38] J. G. Proakis and D. G. Manolakis. *Digital signal processing: principles, algorithms, and applications.* Macmillan, New York, USA, 2nd edition, 1992.

[39] L. R. Rabiner. On the use of autocorrelation analysis for pitch detection. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 25(1):24–33, 1977.

[40] L. R. Rabiner and B. Gold. *Theory and applications of digital signal processing.* Prentice-Hall, England Cliffs, New Jersy, USA, 1975.

[41] B. E. Richardson. Stringed instruments – plucked. In *Encyclopedia of Acoustics*, pages 1627–1634. New York; Chichester: Wiley, 1997.

[42] C. Roads. *The computer music tutorial*, pages 504–520. The MIT Press, Cambridge, Massachusetts, USA, 1995.

[43] J. Sadowsky. The continuous wavelet transform: a tool for signal investigation and understanding. *Johns Hopkins APL Technical Digest*, 15(4):306–318, 1994.

[44] M. R. Schroeder. A new method of measuring reverbration time. *Journal of the Acoustical Society of America*, 37:409–412, 1965.

[45] X. Serra and J. O. Smith. Spectral modelling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decompositions. *Computer Music Journal*, 14(4):12–24, 1990.

[46] W. Shabana. A wavelet-based pitch detector for musical signals. Technical report, Department of Mathematical Sciences, University of Bath, 1999.

[47] J. O. Smith. Music applications of digital waveguides. Technical Report STANM-39, CCRMA, Dept. of Music, Stanford University, California, USA, 1987.

[48] J. O. Smith. Fundamentals of digital filter theory. In J. Strawn, editor, *Digital Audio Signal Processing: An Anthology.* William Kanfmann, Inc., Los Altos, California, 1989. A shortened version appears in The Music Machine, Roads, C.(ed)., MIT Press, 1989.

[49] J. O. Smith. Physical modelling using digital waveguides. *Computer Music Journal*, 16(4):74–91, 1992.

[50] J. O. Smith. Efficient synthesis of stringed musical instruments. In *Proceedings of ICMC, International Computer Music Association, Tokyo, Japan*, pages 64–71, 1993.

[51] J. O. Smith and S. A. Van Duyne. Physical modelling with the 2-D digital waveguide mesh. In *Proceedings of ICMC, International Computer Music Association, Tokyo, Japan*, 1993.

[52] J.O. Smith and X. Serra. Parshl: an analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation. In *Proceedings of the 1987 International Computer Music Conference*, 1987.

[53] G. Strang. Wavelets and dilation equations: a brief introduction. *SIAM Review*, 31(4):614–627, 1989.

[54] G. Strang and T. Nguyen. *Wavelets and filter banks.* Wellesley-Cambridge Press, Boston, 1996.

[55] T. Tolonen. Model-based analysis and resynthesis of acoustic guitar tones. Master's thesis, Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Espoo, Finland, 1998.

[56] T. Tolonen, V. Valimaki, and M. Karjalainen. Evaluation of modern sound synthesis methods. Technical report, Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Espoo, Finland, 1998.

[57] M. Unser. Ten good reasons for using spline wavelets. In *Proc. SPIE v.(3169), Wavelet Applications in Signal and Image Processing V*, pages 422–431, 1997.

[58] V. Valimaki. *Discrete-Time Modeling of Acoustic Tubes using Fractional Delay Filters.* PhD thesis, Helsinki University of Technology, Espoo, Finland, 1995.

[59] V. Valimaki, M. Karjalainen, and Z. Janos. Towards high-quality sound synthesis of the guitar and string instruments. In *Proceedings of the International Computer Music Conference, Tokyo, Japan*, pagès 56–63, 1993.

[60] V. Valimaki, M. Karjalainen, T. I. Laakso, and U. K. Laine. Splitting the unit delay-tools for fractional delay filter design. *IEEE Signal Processing Magazine*, 13(1):30–60, 1996.

[61] V. Valimaki, M. Karjalainen, and T. Tolonen. Plucked string models: from Karplus-Strong algorithm to digital waveguides and beyond. *Computer Music Journal*, 22(3):17–32, 1998.

[62] V. Valimaki and T. Takala. Virtual musical instruments – natural sound using physical models. *Organised Sound*, 1(2):75–86, 1996.

[63] E. H. Wold. Nonlinear parameter estimation of acoustic models. Technical Report UCB/CSD 87/354, Computer Science Division (EECS), University of California, Berkeley, California 94720, 1987.